



ESCUELA SUPERIOR POLITÉCNICA DEL CHIMBORAZO
FACULTAD DE INFORMATICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**“ADAPTACIÓN DE LAS METODOLOGIAS ÁGILES SCRUM Y
EXTREME GAME DEVELOPMENT EN UNA METODOLOGIA
PARA DESARROLLO DE VIDEOJUEGOS EN ANDROID.
CASO PRÁCTICO: DESARROLLO DE UN VIDEOJUEGO”**

TESIS DE GRADO

**PREVIA A LA OBTENCION DEL TÍTULO DE:
INGENIEROS EN SISTEMA INFORMÁTICOS**

**PRESENTADO POR:
GABRIEL ALEXANDER ARMENDÁRIZ BARRENO
MILTON GONZALO SALTOS GUARACA**

**RIOBAMBA – ECUADOR
2013**

AGRADECIMIENTO

A la Escuela Superior Politécnica de Chimborazo por ser la Institución que cristalizó nuestra formación, a sus Autoridades y Docentes, en especial al Ing. Patricio Moreno Director de Tesis y al Ing. Washington Luna Miembro del Tribunal, quienes con su ayuda y colaboración nos permitieron culminar con el desarrollo este trabajo.

DEDICATORIA

Todo el esfuerzo y dedicación que están reflejados en este trabajo dedico a una persona muy especial Andrea y a mis padres Raymundo y Enma por el apoyo incondicional, siendo los pilares fundamentales en mi vida, por su amor, sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, quienes son dignos de ejemplo, trabajo y constancia que siempre me brindan todo el apoyo necesario para alcanzar mis metas y sueños.

Gabriel Armendáriz.

Dedico este trabajo a Dios por ser quien ha estado a mi lado en todo momento dándome las fuerzas necesarias para continuar luchando día tras día y seguir adelante rompiendo todas las barreras que se me presenten.

A mi madre pilar fundamental para terminar este sueño, por haberme educado y soportado mis errores. Gracias por brindarme sus consejos, cultivar e inculcar ese sabio don de la responsabilidad con valores, principios, perseverancia y empeño, y todo ello con una gran dosis de amor y sin pedir nunca nada a cambio.

Milton Saltos.

FIRMAS DE RESPONSABILIDAD

NOMBRE	FIRMAS	FECHA
Ing. Iván Méndez DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
Ing. Raúl Rosero DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS
Ing. Patricio Moreno DIRECTOR DE TESIS
Ing. Washington Luna MIEMBRO DEL TRIBUNAL
Tlgo. Carlos Rodríguez DIRECTOR DEL CENTRO DE DOCUMENTACIÓN
NOTA DE LA TESIS	

“Nosotros GABRIEL ALEXANDER ARMENDÁRIZ BARRENO y MILTON GONZALO SALTOS GUARACA, somos responsables de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado, y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”

GABRIEL ARMENDÁRIZ

MILTON SALTOS

ÍNDICE DE ABREVIATURAS

API	Application Programming Interface o Interfaz de Programación de Aplicaciones
ADT	Android Development Tools
ADB	Android Debug Bridge
BSD	Berkeley Software Distribution
CPU	Unidad central de procesamiento
EDSAC	Electronic Delay Storage Automatic Calculator
GDI	Graphics Device Interface
GPS	Global Positioning System o Sistema de Posicionamiento Global
GSM	Global System for Mobile Communications o Sistema Global para las comunicaciones Móviles
HSPA	High-Speed Packet Access
IDE	Integrated Development Environment o Entorno de Desarrollo Integrado
JSR	Java Specification Request
JDK	Java Development Kit
ME	Micro Edition
MIT	Massachusetts Institute of Technology
NEC	Nippon Electric Company, Limited.

NDK	Native Development Kit
PDP-1	Programmed Data Processor-1
RUP	Proceso Unificado de Rational
SDK	Software Development Kit o Kit de Desarrollo de Software
SGI	Silicon Graphics Incorporated
SGL	Sorcerer Gnu Linux
SMS	Short Message Service o Servicio de Mensajes Cortos
SO	Sistema Operativo.
UML	Lenguaje de Modelado Unificado
USB	Universal Serial Bus o Conductor Universal en Serie
XGD	Extreme Game Development
XP	eXtreme Programming

INDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

ÍNDICE DE ABREVIATURAS

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO I

1. MARCO REFERENCIAL.....	20
1.1. ANTECEDENTES.....	20
1.2. JUSTIFICACIÓN.....	23
1.3. OBJETIVOS.....	25
1.3.1. Objetivo General.....	25
1.3.2. Objetivos Específicos.....	25
1.4. HIPÓTESIS.....	25
1.5. MÉTODOS Y TÉCNICAS.....	25

CAPÍTULO II

2. METODOLOGIAS ÁGILES PARA DESARROLLO DE VIDEOJUEGOS EN LA PLATAFORMA ANDROID.....	27
2.1. Metodologías Agiles para Desarrollo de Videojuegos.....	28

2.1.1. ¿Qué es una Metodología?.....	28
2.1.2. Metodologías Ágiles.....	29
2.1.3. Ventajas de las Metodologías Ágiles.....	31
2.1.4. Beneficios de las Metodologías Ágiles.....	31
2.1.5. Metodología SCRUM.....	32
2.1.5.1. Características.....	33
2.1.5.2. Ventajas.....	33
2.1.5.3. Desventajas.....	34
2.1.5.4. Prácticas.....	34
2.1.5.5. Valores.....	35
2.1.5.6. Roles.....	36
2.1.5.7. Reuniones.....	37
2.1.5.8. Sprint.....	42
2.1.5.9. Fases.....	43
2.1.5.10. Artefactos.....	47
2.1.6. Metodología Extreme Game Development.....	51
2.1.6.1. Características.....	51
2.1.6.2. Ventajas.....	52
2.1.6.3. Desventajas.....	52
2.1.6.4. Prácticas.....	54
2.1.6.5. Valores.....	55
2.1.6.6. Roles.....	56
2.1.6.7. Reuniones.....	58
2.1.6.8. Proceso.....	60

2.1.6.9. Actividades.....	60
2.1.6.10. Fases.....	61
2.1.6.11. Artefactos.....	63
2.2. VIDEOJUEGOS.....	67
2.2.1. Historia.....	67
2.2.2. ¿Qué es un videojuego?.....	70
2.2.3. Principales Géneros.....	71
2.2.4. Características.....	73
2.2.5. Videojuegos 2D.....	74
2.2.6. Partes de un videojuego.....	75
2.2.7. Estructura interna de un videojuego.....	77
2.3. ANDROID.....	78
2.3.1. Arquitectura.....	79
2.3.2. Características.....	83
2.3.3. Ventajas para el Desarrollo de Aplicaciones.....	84
2.3.4. Requisitos que debe cumplir una terminal Android.....	85
2.3.5. La máquina virtual Dalvik.....	86
2.3.6. Entorno de desarrollo de Android.....	87
2.3.7. Aplicaciones.....	88
2.3.8. Estructura de una Aplicación.....	89
2.3.9 Ciclo de vida de las Aplicaciones Android.....	91

CAPÍTULO III

3. ANÁLISIS Y ADAPTACIÓN DE LAS METODOLOGIAS ÁGILES SCRUM Y EXTREME GAME DEVELOPMENT EN UNA METODOLOGÍA PARA

EL DESARROLLO DE VIDEOJUEGOS.....	93
3.1. DETERMINACIÓN DE CRITERIOS PARA EL ANÁLISIS.....	94
3.2. ESCALA DE EVALUACIÓN.....	97
3.3. PROTOTIPO DE VIDEOJUEGO PARA EL ANÁLISIS.....	99
3.3.1. Herramientas utilizadas para el desarrollo del prototipo.....	99
3.4. PROTOTIPO “METODOLOGÍA ÁGIL SCRUM”.....	99
3.4.1. Fase Pre-game.....	99
3.4.2. Fase Game.....	105
3.4.3. Fase Post-game.....	113
3.5. PROTOTIPO “METODOLOGÍA ÁGIL XGD”.....	113
3.5.1. Fase Exploración.....	113
3.5.2. Fase Planificación.....	116
3.5.3. Fase Iteraciones.....	118
3.5.4. Fase Producción.....	124
3.5.5. Fase Mantenimiento.....	125
3.5.6. Fase Muerte del Proyecto.....	125
3.6. RESULTADO DEL ANÁLISIS EN CADA CRITERIO.....	126
3.6.1. Planificación.....	126
3.6.2. Ciclo de Vida del Proyecto.....	129
3.6.3. Calidad.....	133
3.6.4. Herramientas utilizadas en el desarrollo.....	137
3.7. PUNTAJES ALCANZADOS.....	139
3.8. RESULTADO DEL ANÁLISIS.....	142
3.9. VENTAJAS Y DESVENTAJAS.....	143

3.9.1. SCRUM.....	143
3.9.2. XGD.....	145
3.10. ADAPTACIÓN DE LAS METODOLOGÍAS.....	145
3.10.1. Valores.....	146
3.10.2. Prácticas.....	147
3.10.3. Roles.....	149
3.10.4. Reuniones.....	153
3.10.4.1. Reunión Diaria de DAV.....	153
3.10.4.2. Reunión de Planificación de la Iteración.....	154
3.10.4.3. Reunión de Revisión de la Iteración.....	156
3.10.4.4. Reunión de Retrospectiva de la Iteración.....	157
3.10.5. Fases.....	158
3.10.5.1 Fase Pre-juego.....	159
3.10.5.2. Fase Juego.....	160
3.10.5.3. Post-juego.....	162
3.10.6. Artefactos.....	162
3.10.6.1. Historia de Usuario.....	163
3.10.6.2. Prueba de Aceptación.....	164
3.10.6.3. Reserva del Producto.....	166
3.10.6.4. Tarjeta de Tarea.....	169
3.10.6.5. Reserva de Iteración.....	169
3.10.6.6. Burn-Down.....	171
3.10.6.7. Burn-Up.....	172
3.10.6.8. Cartas DAV.....	174

3.10.6.9. Herramientas.....	176
-----------------------------	-----

CAPÍTULO IV

4. DESARROLLO DE UN VIDEOJUEGO.....	179
-------------------------------------	-----

4.1. FASE PRE-JUEGO.....	179
--------------------------	-----

4.1.1 Descripción del videojuego.....	179
---------------------------------------	-----

4.1.2. Aspectos Fundamentales.....	180
------------------------------------	-----

4.1.3. Arquitectura del Videojuego.....	185
---	-----

4.1.4. Diseño Conceptual.....	186
-------------------------------	-----

4.1.5. Herramientas a Utilizarse.....	187
---------------------------------------	-----

4.1.6. Roles.....	187
-------------------	-----

4.1.7. Artefactos.....	188
------------------------	-----

4.2. FASE JUEGO.....	199
----------------------	-----

4.2.1. Iteraciones.....	199
-------------------------	-----

4.3. FASE POST-JUEGO.....	227
---------------------------	-----

4.4. COMPROBACIÓN DE LA HIPÓTESIS.....	227
--	-----

4.4.1 Hipótesis Planteada.....	227
--------------------------------	-----

4.4.2. Tipo de hipótesis.....	227
-------------------------------	-----

4.4.3. Determinación de las variables.....	227
--	-----

4.4.4. Operacionalización Conceptual.....	228
---	-----

4.4.5. Operacionalización Metodológica.....	228
---	-----

4.4.6. Demostración de la Hipótesis.....	230
--	-----

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

TABLA III.I. DETERMINACIÓN DE LOS CRITERIOS PARA EL ANÁLISIS..	93
TABLA III.II. PARÁMETROS DEL CRITERIO “PLANIFICACIÓN” PARA EL ANÁLISIS.....	94
TABLA III.III. PARÁMETROS DEL CRITERIO “CICLO DE VIDA DEL PROYECTO” PARA EL ANÁLISIS.....	94
TABLA III.IV. PARÁMETROS DEL CRITERIO “CALIDAD” PARA ANÁLISIS.....	95
TABLA III.V. PARÁMETROS DEL CRITERIO “HERRAMIENTAS UTILIZADAS EN EL DESARROLLO” PARA EL ANÁLISIS.....	95
TABLA III.VI. ESCALA DE EVALUACIÓN PARA LAS METODOLOGÍAS....	96
TABLA III.VII. PESOS EN EL CRITERIO PARA LOS PARÁMETROS	96
TABLA III.VIII. INTERPRETACIÓN DE RESULTADOS EN CADA METODOLOGÍA.....	97
TABLA III.IX. CALIFICACIÓN PARA LAS METODOLOGÍAS.....	98
TABLA III.X. ROLES ASIGNADOS PARA EL DESARROLLO DEL PROTOTIPO CON SCRUM.....	99
TABLA III.XI. ELEMENTOS DEL PROTOTIPO DE VIDEOJUEGO EN SCRUM.....	100
TABLA III.XII. USER STORIES DEL PROTOTIPO DE VIDEOJUEGO EN SCRUM.....	100
TABLA III.XIII. PRODUCT BACKLOG DEL PROTOTIPO DE VIDEOJUEGO EN SCRUM.....	101
TABLA III.XIV. USER STORIES A IMPLEMENTARSE EN EL PRIMER SPRINT.....	104
TABLA III.XV. USER STORIES A IMPLEMENTARSE EN EL SEGUNDO SPRINT.....	108
TABLA III.XVI. ROLES ASIGNADOS PARA EL DESARROLLO DEL PROTOTIPO CON XGD.....	112
TABLA III.XVII. USER STORIES PRIMERA ENTREGA “XGD”.....	113
TABLA III.XVIII. USER STORIES SEGUNDA ENTREGA “XGD”.....	114
TABLA III.XIX. PRIORIDADES Y ESFUERZO DE LOS USER STORIES PARA LA PRIMERA ENTREGA.....	115
TABLA III.XX. PRIORIDADES Y ESFUERZO DE LOS USER STORIES PARA LA SEGUNDA ENTREGA.....	116
TABLA III.XXI. FECHAS DE ENTREGA DE LAS ITERACIONES DE XGD..	117
TABLA III.XXII. DETALLE USER STORY 01 “XGD”.....	117
TABLA III.XXIII. PRUEBA DE ACEPTACIÓN USER STORY 01 “XGD”.....	118
TABLA III.XXIV. TASK CARD DEL USER STORY 01-01 “XGD”.....	118
TABLA III.XXV. TASK CARD DEL USER STORY 01-02 “XGD”.....	119

TABLA III.XXVI. CARD CRC DEL USER STORY 01 “XGD”.....	119
TABLA III.XXVII. PESOS DE LOS PARÁMETROS PARA LAS DOS METODOLOGÍAS.....	124
TABLA III.XXVIII. CRITERIO “PLANIFICACIÓN” METODOLOGÍA SCRUM.....	125
TABLA III.XXIX. CRITERIO “PLANIFICACIÓN” METODOLOGÍA XGD...	125
TABLA III.XXX. RESULTADOS DEL CRITERIO “PLANIFICACIÓN”.....	126
TABLA III.XXXI. PESOS DE LOS PARÁMETROS PARA LAS DOS METODOLOGÍAS.....	128
TABLA III.XXXII. CRITERIO “CICLO DE VIDA DEL PROYECTO” METODOLOGÍA SCRUM.....	128
TABLA III.XXXIII. CRITERIO “CICLO DE VIDA DEL PROYECTO” METODOLOGÍA XGD.....	129
TABLA III.XXXIV. RESULTADOS DEL CRITERIO “CICLO DE VIDA DEL PROYECTO”.....	130
TABLA III.XXXV. PESOS DE LOS PARÁMETROS PARA LAS DOS METODOLOGÍAS.....	132
TABLA III.XXXVI. CRITERIO “CALIDAD” METODOLOGÍA SCRUM.....	132
TABLA III.XXXVII. CRITERIO “CALIDAD” METODOLOGÍA XGD.....	132
TABLA III.XXXVIII. RESULTADOS DEL CRITERIO “CALIDAD”.....	133
TABLA III.XXXIX. PESOS DE LOS PARÁMETROS PARA LAS DOS METODOLOGÍAS.....	135
TABLA III.XL. CRITERIO “HERRAMIENTAS UTILIZADAS EN EL DESARROLLO” METODOLOGÍA SCRUM.....	135
TABLA III.XLI. CRITERIO “HERRAMIENTAS UTILIZADAS EN EL DESARROLLO” METODOLOGÍA XGD.....	136
TABLA III.XLII. RESULTADOS DEL CRITERIO “HERRAMIENTAS UTILIZADAS EN EL DESARROLLO”.....	137
TABLA III.XLIII. TABLA GENERAL DE RESULTADOS.....	138
TABLA III.XLIV. CALIFICACIÓN FINAL DE LAS METODOLOGÍAS SCRUM Y XGD.....	141
TABLA III.XLV. HERRAMIENTAS QUE PUEDEN SER UTILIZADAS POR DAV.....	174
TABLA IV.XLVI. DISEÑOS PRELIMINARES DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”.....	178
TABLA IV.XLVII. ASPECTOS DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”.....	180
TABLA IV.XLVIII. DISEÑOS PRELIMINARES DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”.....	180
TABLA III.XLIX. ROLES ASIGNADOS PARA EL DESARROLLO DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”.....	186
TABLA III.L. HISTORIAS DE USUARIO DEL VIDEOJUEGO DAV.....	186

TABLA IV.LI. RESERVA DEL PRODUCTO DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”	191
TABLA IV.LII. FECHAS DE ENTREGA DE CADA ITERACIÓN “EL TREN DEL CONOCIMIENTO”	197
TABLA IV.LIII. HISTORIAS DE USUARIO A IMPLEMENTARSE EN LA ITERACIÓN 1	198
TABLA IV.LIV. HISTORIAS DE USUARIO A IMPLEMENTARSE EN LA ITERACIÓN 2	203
TABLA N° IV.LV. HISTORIAS DE USUARIO A IMPLEMENTARSE EN LA ITERACIÓN 3	207
TABLA IV.LVI. HISTORIAS DE USUARIO A IMPLEMENTARSE EN LA ITERACIÓN 4	211
TABLA IV.LVII. HISTORIAS DE USUARIO A IMPLEMENTARSE EN LA ITERACIÓN 5	216
TABLA IV.LVIII. HISTORIAS DE USUARIO A IMPLEMENTARSE EN LA ITERACIÓN 6	220
TABLA IV.LIX. OPERACIONALIZACIÓN CONCEPTUAL	226
TABLA IV.LX. OPERACIONALIZACIÓN METODOLÓGICA	226
TABLA IV.LXI. DEMOSTRACIÓN DE LA HIPÓTESIS	228

ÍNDICE DE FIGURAS

FIGURA II.1. METODOLOGÍA.....	29
FIGURA II.2. ESQUEMA DE TRABAJO SCRUM.....	32
FIGURA II.3. CICLO DE SPRINTS DE SCRUM.....	42
FIGURA II.4. FASES DE SCRUM.....	43
FIGURA II.5. USER STORY DE SCRUM.....	47
FIGURA II.6. PRODUCT BACKLOG.....	48
FIGURA II.7. SPRINT BACKLOG.....	49
FIGURA II.8. GRÁFICO DE BURN-DOWN.....	49
FIGURA II.9. GRÁFICO DE BURN-UP.....	50
FIGURA II.10. FASES DE UN PROYECTO EN XGD.....	60
FIGURA II.11. USER STORY XGD.....	63
FIGURA II.12. CASO DE PRUEBA DE ACEPTACIÓN “XGD”.....	64
FIGURA II.13. TASK CARD “XGD”.....	64
FIGURA II.14. CARD CRC “XGD”.....	65
FIGURA II.15. EL PRIMER VIDEOJUEGO ”NOUGHT AND CROSSES”.....	67
FIGURA II.16. VIDEOJUEGOS.....	70
FIGURA II.17. PARTES DE UN VIDEOJUEGO.....	74
FIGURA II.18. ESTRUCTURA DE UN VIDEOJUEGO.....	76
FIGURA II.19. LOGO ANDROID.....	77
FIGURA II.20. ARQUITECTURA INTERNA DEL CONJUNTO DE SOFTWARE ANDROID.....	78
FIGURA II.21. FORMATO DE UN FICHERO .DEX.....	85
FIGURA II.22. ENTORNO DE DESARROLLO DE ANDROID.....	86
FIGURA II.23. CICLO DE VIDA DE UN OBJETO ACTIVITY.....	91
FIGURA III.24. DISEÑO CONCEPTUAL DEL VIDEOJUEGO.....	99
FIGURA III.25. SPRINT BACKLOG DEL PRIMER SPRINT.....	105
FIGURA III.26. GRÁFICO DE ESFUERZO DEL PRIMER SPRINT.....	106
FIGURA III.27. GRÁFICO DE TAREAS DEL PRIMER SPRINT.....	106
FIGURA III.28. FUNCIONALIDAD DEL PROTOTIPO AL FINALIZAR EL PRIMER SPRINT.....	107
FIGURA III.29. SPRINT BACKLOG DE LA SEGUNDA ITERACIÓN.....	109
FIGURA III.30. GRÁFICO DE ESFUERZO DEL SEGUNDO SPRINT.....	109
FIGURA III.31. GRÁFICO DE TAREAS DEL SEGUNDO SPRINT.....	110
FIGURA III.32. FUNCIONALIDAD DEL PROTOTIPO AL FINALIZAR EL SEGUNDO Y ÚLTIMO SPRINT.....	111
FIGURA III.33. BOSQUEJO OBTENIDO PARA LA PRIMEA ENTREGA “XGD”.....	113
FIGURA III.34. BOSQUEJO OBTENIDO PARA LA SEGUNDA ENTREGA “XGD”.....	114

FIGURA III.35. REFACTORIZACIÓN DEL CÓDIGO.....	120
FIGURA III.36. SEGUIMIENTO DEL ESTADO DE LAS TAREAS DE LA PRIMERA ITERACIÓN “XGD”.....	121
FIGURA III.37. SEGUIMIENTO DEL ESTADO DE LAS TAREAS DE LA SEGUNDA ITERACIÓN “XGD”.....	122
FIGURA III.38. PARTE FUNCIONAL PRIMERA ITERACIÓN “XGD”.....	123
FIGURA III.39. PARTE FUNCIONAL SEGUNDA ITERACIÓN “XGD”.....	123
FIGURA III.40. CRITERIO “PLANIFICACIÓN”.....	127
FIGURA III.41. CRITERIO “CICLO DE VIDA DEL PROYECTO”.....	130
FIGURA III.42. CRITERIO “CALIDAD”.....	134
FIGURA III.43. CRITERIO “HERRAMIENTAS UTILIZADAS EN EL DESARROLLO”.....	137
FIGURA III.44. DIAGRAMA GENERAL DE RESULTADOS.....	139
FIGURA III.45. RESULTADO FINAL.....	140
FIGURA III.46. ROLES DE DAV.....	147
FIGURA III.47. PROCESO DE DAV.....	159
FIGURA III.48. HISTORIA DE USUARIO DE DAV.....	161
FIGURA III.49. PRUEBA DE ACEPTACIÓN DE DAV.....	163
FIGURA III.50. RESERVA DEL PRODUCTO.....	165
FIGURA III.51. TAREA DE INGENIERÍA DAV.....	167
FIGURA III.52. RESERVA DE ITERACIÓN.....	168
FIGURA III.53. GRÁFICO BURN-DOWN DE DAV.....	170
FIGURA III.54. GRÁFICO DE BURN-UP.....	171
FIGURA III.55. CARTAS DAV.....	173
FIGURA IV.56. ARQUITECTURA DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”.....	183
FIGURA IV.57. DISEÑO CONCEPTUAL DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”.....	184
FIGURA IV.58. PRUEBA DE ACEPTACIÓN “PA01”.....	190
FIGURA IV.59. TARJETA DE TAREA “T01”.....	199
FIGURA IV.60. RESERVA DE ITERACIÓN 1.....	199
FIGURA IV.61. GRÁFICO BURN-DOWN DE LA ITERACIÓN 1.....	200
FIGURA IV.62. TAREAS PENDIENTES DE LA ITERACIÓN 1.....	200
FIGURA IV.63. REFACTORIZACIÓN DEL CÓDIGO DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”.....	201
FIGURA IV.64. ENTREGA FUNCIONAL DE LA ITERACIÓN 1.....	202
FIGURA IV.65. TARJETA DE TAREA “T07”.....	204
FIGURA IV.66. RESERVA DE ITERACIÓN 2.....	204
FIGURA IV.67. GRÁFICO BURN-DOWN DE LA ITERACIÓN 2.....	205
FIGURA IV.68. TAREAS PENDIENTES DE LA ITERACIÓN 2.....	205
FIGURA IV.69. ENTREGA FUNCIONAL DE LA ITERACIÓN 2.....	206
FIGURA IV.70. TARJETA DE TAREA “T14”.....	208

FIGURA IV.71. RESERVA DE ITERACIÓN 3.....	209
FIGURA IV.72. GRÁFICO BURN-DOWN DE LA ITERACIÓN 3.....	209
FIGURA IV.73. TAREAS PENDIENTES DE LA ITERACIÓN 3.....	210
FIGURA IV.74. ENTREGA FUNCIONAL DE LA ITERACIÓN 3.....	211
FIGURA IV.75. TARJETA DE TAREA “T22”.....	213
FIGURA IV.76. RESERVA DE ITERACIÓN 4.....	213
FIGURA IV.77. GRÁFICO BURN-DOWN DE LA ITERACIÓN 4.....	214
FIGURA IV.78. TAREAS PENDIENTES DE LA ITERACIÓN 4.....	214
FIGURA IV.79. ENTREGA FUNCIONAL DE LA ITERACIÓN 4.....	215
FIGURA IV.80. TARJETA DE TAREA “T29”.....	217
FIGURA IV.81. RESERVA DE ITERACIÓN 4.....	218
FIGURA IV.82. GRÁFICO BURN-DOWN DE LA ITERACIÓN 5.....	218
FIGURA IV.83. TAREAS PENDIENTES DE LA ITERACIÓN 5.....	219
FIGURA IV.84. ENTREGA FUNCIONAL DE LA ITERACIÓN 5.....	220
FIGURA IV.85. TARJETA DE TAREA “T37”.....	222
FIGURA IV.86. RESERVA DE ITERACIÓN 6.....	222
FIGURA IV.87. GRÁFICO BURN-DOWN DE LA ITERACIÓN 6.....	223
FIGURA IV.88. TAREAS PENDIENTES DE LA ITERACIÓN 6.....	223
FIGURA IV.89. ENTREGA FUNCIONAL DE LA ITERACIÓN 6.....	224

INTRODUCCIÓN

Los videojuegos se encuentran cada vez más presentes en nuestra vida diaria y en una sociedad orientada al consumo tecnológico con una tendencia a acrecentarse en cualquier dispositivo con capacidades multimedia (celulares, tablets, televisión digital, etc.), abarcando no solamente a niños, sino también a personas de todas las edades y sexos.

El desarrollo de videojuegos, por su parte, no tiene estandarizado una metodología para el desarrollo de sus productos, es por eso que en muchas ocasiones los costos se van por encima de lo planificado, los tiempos de entrega se alargan y los desarrolladores cargan con excesos de trabajo. Por tal motivo se pretende desarrollar esta propuesta adaptando las metodologías ágiles Scrum y Extreme Game Development en una nueva metodología para desarrollo de videojuegos.

Esta metodología pretende facilitar el desarrollo de videojuegos en la plataforma Android de forma técnica y sencilla.

Para realizar este trabajo de investigación se ha decidido dividir el documento en cuatro capítulos los cuales se detallan a continuación:

- En el **capítulo I**, se presenta los antecedentes, justificación, objetivos a cumplir y la hipótesis de la investigación.
- En el **capítulo II**, se realiza el Marco Teórico para las metodologías ágiles Scrum y Extreme Game Development, Videojuegos y la Plataforma Android.
- En el **capítulo III**, se enfoca al análisis y adaptación entre las metodologías ágiles Scrum y Extreme Game Development para desarrollo de videojuegos en Android.
- En el **capítulo IV**, se enfoca al desarrollo de un videojuego en 2 Dimensiones en la plataforma Android aplicando la nueva metodología adaptada DAV.

Finalmente, este trabajo concluye emitiendo las conclusiones a las que se llegó luego de realizar la investigación, así como las debidas recomendaciones para la utilización de la nueva metodología adaptada DAV en el desarrollo de videojuegos en Android.

CAPÍTULO I

1. MARCO REFERENCIAL

En este capítulo se explica la problemática del trabajo, en el cual se encuentra: los antecedentes, la justificación, los objetivos a cumplir y la hipótesis.

1.1. ANTECEDENTES

Los videojuegos han sufrido una continua evolución hasta llegar a las sofisticadas máquinas que en la actualidad tenemos a nuestro alcance como son los dispositivos móviles.

La historia de los videojuegos tiene origen en la década de 1940 pero la verdadera revolución de los juegos en red surgió en 1993 con la creación de la World Wide Web. Los usuarios tenían la posibilidad de acceder gratuitamente a versiones reducidas de videojuegos para ordenador con fines básicamente promocionales, como es el caso de la primera entrega del Doom. Además la rápida difusión de Internet como medio de

entretenimiento facilitó la mejora de las tecnologías para la conexión en red de usuarios y su acercamiento a la sociedad.

En 2001, las principales compañías móviles anunciaron el Foro de Interoperabilidad de Juegos Móviles que trabaja para definir una especificación de interoperabilidad de juegos móviles para los servidores basados en redes. Esto permitiría que los desarrolladores de juegos produzcan juegos móviles que puedan distribuirse a través de diferentes servidores de juegos y redes inalámbricas y puedan ser ejecutados en diferentes dispositivos móviles.

Un videojuego es un software creado para el entretenimiento en general y basado en la interacción entre una o varias personas por medio de un controlador y un aparato electrónico que ejecuta dicho videojuego, este dispositivo electrónico puede ser una computadora, una videoconsola, un dispositivo portátil como por ejemplo un teléfono móvil los cuales son conocidos como "plataformas". Aunque, usualmente el término "video" en la palabra "videojuego" se refiere en sí a un visualizador de gráficos rasterizados, hoy en día se utiliza para hacer uso de cualquier tipo de visualizador, se entiende por videojuego todo tipo de juego digital interactivo, con independencia de su soporte.

En muchos casos, los videojuegos recrean entornos y situaciones virtuales en los que el video jugador puede controlar a uno o varios personajes, para conseguir uno o varios objetivos por medio de reglas determinadas.

Existen multitud de formas de interactuar con un videojuego, aunque se podría decir que siempre es necesario un dispositivo externo, esto no es del todo correcto, debido a que existen consolas portátiles, que permiten jugar mediante su pantalla táctil

Una de las razones para comenzar a desarrollar primero videojuegos 2D, es que los conceptos involucrados son mucho más simples y fáciles de asimilar.

El desarrollo de videojuegos está muy relacionado con el desarrollo de software tradicional, por eso hoy en día se está implementando un proceso similar para lograr un trabajo más organizado.

La industria de los videojuegos en general no tiene estandarizado una metodología para el desarrollo de sus productos, debido a esta situación han sido muchas las empresas que se han dedicado a plantearse una metodología que permita un desarrollo adecuado y estandarizado.

Estas metodologías no son más que adaptaciones de las ya existente para el desarrollo de software tradicional solo que ajustadas a las necesidades del proceso de desarrollo de un videojuego, en particular se registran casos de éxito con adaptaciones de Scrum y XP

La metodología Scrum fue desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos años, es una metodología ágil para gerenciar y controlar el desarrollo de software de un producto en forma iterativa e incremental.

Una de sus características es que no indica prácticas específicas a seguir durante el desarrollo, lo que brinda flexibilidad y permite ajustar el proceso a la realidad y forma de trabajo de cada proyecto, así como a los diferentes requerimientos de los clientes. Existen casos de empresas en la industria que logran adaptar esta metodología para videojuegos y les reporta beneficios, por ejemplo High Moon Studios, Large Animal Games, Crytek, Relic, DICE y Nokia.

Según la descripción que realiza Ken Schwaber, para videojuegos Scrum se estructura en tres fases denominadas pre-game, game y post-game.

Así mismo la metodología XGD, fue descrita en 2003 por Thomas Demachy a través de un documento presentado al portal Gamasutra.

XGD se define como una metodología ágil para la producción de videojuegos, es una adaptación de XP para videojuegos, una práctica muy común en el desarrollo de software ágil.

El XGD fue desarrollado por la empresa para la que trabajó en el Demachy Time, Tito Estudio Interactivo, que luego desarrolló videojuegos.

Según Demachy, muchos equipos de desarrollo de videojuegos han utilizado el Extreme Programming como metodología estándar. Sin embargo, XP tiene una limitación

importante cuando se aplica al universo de los videojuegos: su creación principalmente para satisfacer las necesidades de los programadores. Desde un proyecto de Juego electrónico involucra a personas de diversas disciplinas, por ello vieron la necesidad de la adaptación de XP por lo que podría absorber esta peculiaridad de los proyectos de videojuego, la electrónica. Como resultado de esta adaptación, se produjo el desarrollo del juego extremo (XGD).

Android es un sistema operativo móvil de código libre, gratuito y multiplataforma basado en el núcleo de Linux, que conjuntamente con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005, actualmente el desarrollo de este sistema operativo está a cargo de la Open Handset Alliance, la cual es liderada por Google.

En la actualidad Android ofrece un conjunto de herramientas de desarrollo disponible para ordenadores con sistema operativo Windows Linux y Mac al igual que iPhone OS. Las aplicaciones para Android pueden escribirse tanto en lenguaje Java como en C/C++, claramente es un medio para poder desarrollar juegos y distribuirlos fácilmente a través de Web Stores, la tienda de aplicaciones Android Market y directamente a través de los ficheros de aplicaciones Android de Google, además existe documentación abundante y actualizada para facilitar la programación como también información y tutoriales.

1.2. JUSTIFICACIÓN

El uso de las metodologías ágiles Scrum y Extreme Game Development en la creación de videojuegos permitirá tener una mayor visión y control del avance de un proyecto, tanto al cliente como a los desarrolladores. Esto se debe a que se pueden determinar nuevas estrategias, iteración por iteración, para lograr llegar en tiempo y forma a los plazos requeridos, involucran a todo el equipo en las decisiones, lo que logra compromiso y motivación.

Al ser metodologías iterativas e incrementales se obtienen versiones jugables del producto en intervalos regulares de tiempo. Esto facilita una visión temprana del

resultado final del juego, lo cual reduce la probabilidad de cambios de requerimientos en forma tardía y brinda una mayor retroalimentación del cliente.

Así también se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto.

Es importante utilizar metodologías ágiles para el desarrollo de videojuegos debido a que nos ayudan a optimizar costos, tiempos de entrega y no hay exceso de trabajo asignado a los desarrolladores.

Se ha escogido Android porque se puede instalar prácticamente en todo tipo de dispositivos, sean móviles o portátiles, esto hace que Android siempre esté presente en los terminales más potentes del mercado debido a que este está liberado con licencia Apache y código abierto, esto lo convierte en un sistema operativo totalmente libre para que un desarrollador no solo pueda modificar su código sino también mejorarlo del mismo modo Android da completa libertad al propietario de un terminal a instalar lo que le convenga, sea desde Android Market como un ejecutable aparte (apk) no limitando la libertad del usuario ni imponiendo software propietario para poder instalar música, archivos, documentos, entre otros.

También se puede decir que Android es libertad en todos los aspectos permitiendo de esta manera que todas las operadoras puedan disfrutar de sus aplicaciones, por el hecho de que Android puede ser instalado en teléfonos de cualquier fabricante o incluso en otros dispositivos, esto permite poder disfrutar de una amplia gama de terminales de diferentes precios y gamas para determinados poderes adquisitivos sin tener que forzar o limitar un sistema operativo o terminal a determinadas clases dando la opción de que toda persona pueda adquirir el terminal que más le guste así como de diferentes características o costos.

Una vez adaptado las metodologías ágiles Scrum y Extreme Game Development en una metodología, se realizará un videojuego, el cual será desarrollado en Android.

El videojuego consistirá en ir conociendo los Cantones de la Provincia de Chimborazo según vaya avanzando los niveles.

Mediante este videojuego se pretende demostrar la importancia de seguir una metodología en su desarrollo, la cual permite optimizar costo, tiempo y trabajo del programador.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Adaptar las metodologías ágiles Scrum y Extreme Game Development en una metodología para el desarrollo de videojuegos en Android y aplicarlo en la realización de un videojuego.

1.3.2. OBJETIVOS ESPECIFICOS

- ✓ Analizar las metodologías ágiles Scrum y Extreme Game Development para el desarrollo de videojuegos en Android.
- ✓ Adaptar las metodologías ágiles Scrum y Extreme Game Development en una metodología para el desarrollo de videojuegos.
- ✓ Desarrollar un videojuego en Android utilizando la metodología adaptada.

1.4. HIPÓTESIS

La adaptación de una metodología en el desarrollo de videojuegos, facilitará la creación de los juegos en la plataforma Android de forma técnica y sencilla.

1.5. MÉTODOS Y TÉCNICAS

Método Científico

Método utilizado con el fin de alcanzar conocimientos válidos sobre las Metodologías Ágiles Scrum y Extreme Game Development en el desarrollo de videojuegos.

Método Deductivo

Este método será utilizado en el análisis de las Metodologías Agiles Scrum y Extreme Game Development en el desarrollo de videojuegos; debido a que se estudiarán desde sus definiciones, características, ventajas, desventajas, roles y fases que posee cada una, buscando los beneficios más relevantes de las mismas.

Método Comparativo

Permitirá comparar y seleccionar los beneficios de calidad que proporcionan las Metodologías Agiles Scrum y Extreme Game Development con el fin de adaptar en una nueva metodología para desarrollo de videojuegos.

TÉCNICAS

Brindarán apoyo a la recolección de la información sobre las Metodologías Agiles Scrum y Extreme Game Development en el desarrollo de videojuegos. Las técnicas empleadas son:

- Revisión de Artículos Científicos.
- Observación.
- Pruebas.

CAPÍTULO II

2. METODOLOGIAS ÁGILES PARA DESARROLLO DE VIDEOJUEGOS EN LA PLATAFORMA ANDROID

El presente capítulo menciona las definiciones conceptuales relacionadas con las metodologías de desarrollo ágil Scrum y Extreme Game Development, permitiendo conocer las características, ventajas, desventajas, practicas, valores, roles, reuniones, fases y artefactos que posee cada una de ellas, así como también conceptos involucrados con los videojuegos y la plataforma Android.

2.1. Metodologías Ágiles para Desarrollo de Videojuegos

El desarrollo de videojuegos está relacionado con el desarrollo de software tradicional, por lo que debería implementarse un proceso similar para lograr un trabajo más organizado. La industria de los videojuegos en general no tiene estandarizado una metodología para el desarrollo de sus productos, es por eso que en muchas ocasiones los costos superan lo planificado, los tiempos de entrega se alargan y los desarrolladores son acumulados con excesivo trabajo. Debido a esta situación han sido muchas las empresas que se han dedicado a plantearse una metodología que permita un desarrollo adecuado y estandarizado; estas metodologías no son más que adaptaciones de las ya existentes para el desarrollo de software tradicional, solo que ajustadas a las especificidades del proceso de desarrollo de un videojuego.

La tendencia a utilizar metodologías ágiles para videojuegos tomó fuerza en los últimos años, por existir varios casos de empresas en la industria que logran adaptar estas metodologías. A pesar de esto, ninguna de estas adaptaciones está especificada formal y públicamente.

2.1.1. ¿Qué es una Metodología?

Una metodología es un conjunto de etapas formalmente estructuradas, de manera que brinden a los interesados los siguientes parámetros de acción en el desarrollo de sus proyectos: plan general y detallado, tareas y acciones, tiempos, aseguramiento de la calidad, involucrados, etapas, revisiones de avance, responsables, recursos requeridos, entre otros [1].

Una metodología está compuesta por:

- Cómo dividir un proyecto en etapas.
- Qué tareas se llevan a cabo en cada etapa.
- Qué restricciones deben aplicarse.
- Qué técnicas y herramientas se emplean.
- Cómo se controla y gestiona un proyecto.

¹ http://www2.rhernando.net/modules/tutorials/doc/ing/met_soft.html

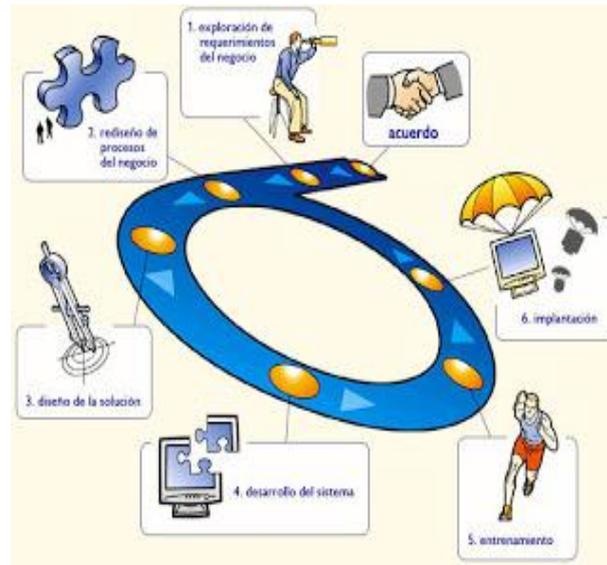


Figura II.1. Metodología.

Fuente: <http://proyectotecnologicodeyadiramamani.blogspot.com/2010/11/metodologia.html>

2.1.2. Metodologías Ágiles

Son metodologías efectivas para modelar y documentar un proyecto de software, tienen una colección de valores, principios y prácticas para modelar software, que pueden ser aplicados de manera simple y ligera [2]. Estas metodologías ofrecen oportunidades para evaluar la dirección de un proyecto durante todo el ciclo de vida de desarrollo. Esto se logra a través de cadencias regulares de trabajo, conocidos como los sprints o iteraciones, al final de las cuales los equipos deben presentar un incremento del producto potencialmente entregable.

Las metodologías ágiles tienen varios principios que las diferencian sobre las metodologías tradicionales reflejados en el Manifiesto Ágil [3] que enuncia cuatro valores que son:

1. **Los individuos y sus relaciones sobre las personas y los procesos.-** Con este principio se hace manifiesto el énfasis de esta metodología sobre las personas, debido a que ellas son de las que depende el éxito o el fracaso de un proyecto, es a las que se les debe motivar.

² <http://www.bi-la.com/profiles/blogs/metodologia-agil-introduccion>

³ [http://msdn.microsoft.com/es-es/library/vstudio/dd997578\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/vstudio/dd997578(v=vs.100).aspx)

2. **Un Software funcional, que trabaje sobre la documentación más completa.-**
Este principio trata de decir que lo más importante es que el software trabaje, cumpla con las necesidades de negocio, no hacer de la documentación un fin en sí mismo, debido a que esta es solo para dar soporte, no es el objetivo primario del desarrollo, existen situaciones en donde incluso la documentación podría ser innecesaria, por ejemplo una pequeña aplicación emergente, que una vez pasada la emergencia, esta aplicación desaparece, el cargar de documentación de requerimientos, arquitectura, testeo etc. Podría considerarse de sobra, sin embargo eso no quiere decir que no es necesaria la documentación, esta debe existir pero solo la suficiente.
3. **Colaboración del cliente sobre el contrato de negocio.-** Se trata de colaborar con el cliente el mayor tiempo no de luchar con el sobre un contrato minucioso, esto puede ser difícil debido a que los clientes no están acostumbrados, ellos están acostumbrados a trabajar sobre un contrato con el que puedan defenderse si las cosas van mal.
4. **Ser capaz de responder a los cambios y no obsesionarse sobre el seguimiento de un plan.-** Es tener la capacidad de adaptación, no decir no a los cambios, aceptar las sugerencias de los usuarios, sin por eso hacer un lado la planificación.

Estos valores han dado lugar a doce principios que son:

1. La satisfacción del cliente.
2. Bienvenida a los cambios que puedan ocurrir.
3. Entregar regularmente software que trabaje.
4. Gente de negocios y desarrolladores trabajan diariamente en conjunto.
5. Construcción de proyectos alrededor de individuos motivados para esto.
6. Las comunicaciones cara a cara son las mejores.
7. Software que trabaje es la mejor medida del progreso.
8. Atención continua a la excelencia y al buen diseño.
9. Promover el desarrollo sostenible.
10. Simplicidad.
11. Las mejores arquitecturas, requerimientos y diseños emergen de equipos auto-organizados.

12. Introspección, los equipos deben regularmente hacerse una revisión hacia sí mismos y sus procesos para intentar mejorar.

2.1.3. Ventajas de las Metodologías Ágiles

Las metodologías ágiles presentan diversas ventajas como:

- ✓ Rápida respuesta a cambios de requisitos a lo largo del desarrollo.
- ✓ Entrega continua y en plazos cortos de software funcional.
- ✓ Trabajo conjunto entre el cliente y el equipo de desarrollo.
- ✓ Minimiza los costos frente a cambios.
- ✓ Importancia de la simplicidad, al eliminar el trabajo innecesario.
- ✓ Mejora continua de los procesos y del equipo de desarrollo.
- ✓ Evita malentendidos de requerimientos entre el cliente y el equipo.
- ✓ El equipo de desarrollo no malgasta el tiempo y dinero del cliente desarrollando soluciones innecesariamente generales y complejas que en realidad no son un requisito del cliente.
- ✓ Cada componente del producto final ha sido probado y satisface los requerimientos.

2.1.4. Beneficios de las Metodologías Ágiles

Estos son los principales beneficios que reportan los casos de éxito al utilizar metodologías ágiles en el desarrollo de videojuegos:

1. Al ser metodologías iterativas e incrementales se obtienen versiones jugables del producto en intervalos regulares de tiempo. Esto facilita una visión temprana del resultado final del juego, lo cual reduce la probabilidad de cambios de requerimientos en forma tardía y brinda una mayor retroalimentación del cliente.
2. Permiten tener una mayor visión y control del avance del proyecto, tanto al cliente como a los desarrolladores. Esto se debe a que se pueden determinar nuevas estrategias, iteración por iteración, para lograr llegar en tiempo y forma a los plazos requeridos.
3. Involucran a todo el equipo en las decisiones, lo que logra compromiso y motivación.

2.1.5. Metodología SCRUM

Fue desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos años, es una metodología ágil para gerenciar y controlar el desarrollo de software de un producto en forma iterativa e incremental, que puede ser usada para manejar el desarrollo de productos complejos de software, ha sido usada desde proyectos simples hasta proyectos de cambios estructurales completos en las empresas para sus negocios [4].

Scrum incrementa significativamente la productividad y reduce el tiempo de espera para ver los beneficios así como facilitar la adaptación de los sistemas desarrollados.

El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o XP [5].

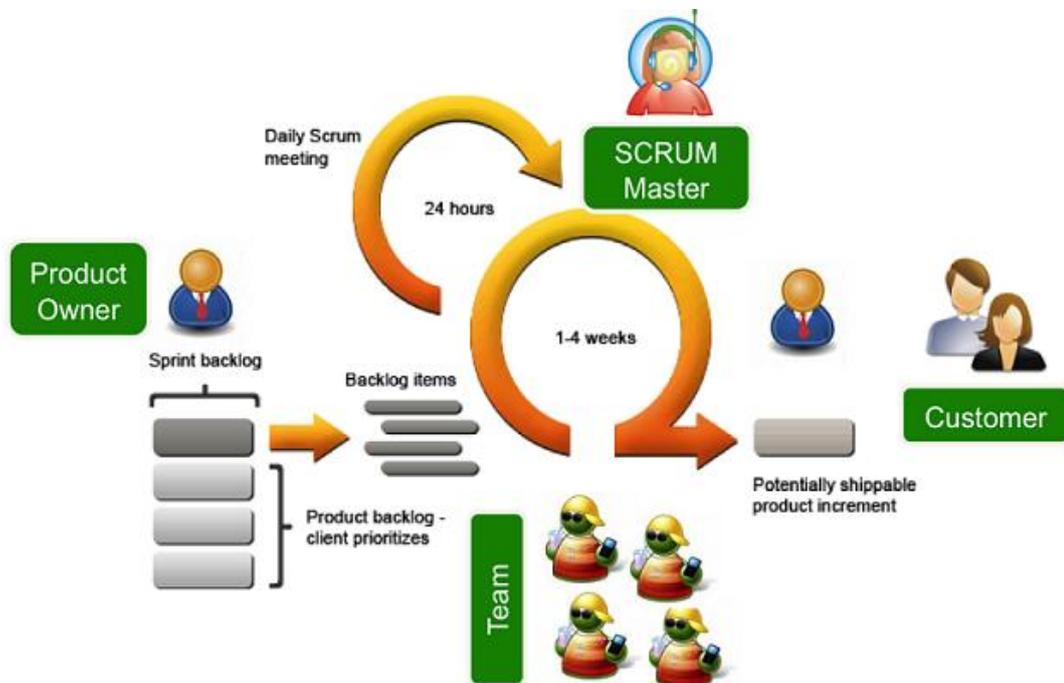


Figura II.2. Metodología SCRUM.

Fuente: <http://eliana-ha.blogspot.com/2010/11/metodologia-scrum.html>

⁴ <http://scrum.org/Resources/What-is-Scrum>

⁵ <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>

En Scrum, el equipo se focaliza en una única cosa: construir software de calidad. Por el otro lado, la gestión de un proyecto Scrum se focaliza en definir cuáles son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo.

2.1.5.1. Características

Scrum es un marco de trabajo ágil que se basa en la iteración y entrega de incrementales de desarrollo de un producto o servicio. Posee las siguientes características [6]:

- Su prioridad es la satisfacción del cliente, que se da con la continua interacción entre éste y el equipo de desarrolladores.
- Se aceptan requisitos cambiantes.
- Enfocado a conseguir pequeños incrementos de software completamente funcionales.
- Es un modo de desarrollo adaptable, antes que predictivo.
- Orientado a las personas, más que a los procesos.
- Emplea el modelo de construcción incremental basado en iteraciones y revisiones.
- Equipos auto-organizado
- Alta flexibilidad.

2.1.5.2. Ventajas

- ✓ Incremento en la productividad.
- ✓ Mejoras constantes.
- ✓ El producto total se convierte en una serie de pequeños pedazos manejables.
- ✓ Existe un progreso, inclusive si los requerimientos no están bien definidos.
- ✓ Todo es visible para todos.
- ✓ Existe una gran comunicación en el equipo.
- ✓ El equipo comparte los éxitos desde el principio hasta el final.
- ✓ El cliente se mantiene informado en cada mejora del producto.

⁶ PALACIO JUAN, *Flexibilidad con Scrum*, 2008

- ✓ Entrega de un producto funcional (jugable) al finalizar cada sprint.
- ✓ Posibilidad de ajustar la funcionalidad en base a las exigencias de los jugadores.
- ✓ Visualización del videojuego día a día.
- ✓ Alcance acotado y viable.
- ✓ Equipos integrados y comprometidos con el desarrollo del videojuego, toda vez que ellos definieron el alcance y se auto administran.
- ✓ Capacidad para aceptar modificaciones sobre la marcha sin influir en el desarrollo.

2.1.5.3. Desventajas

- ✓ Prioridades a funcionalidades del videojuego gracias al Product Backlog.
- ✓ No genera toda la evidencia o documentación de otras metodologías.
- ✓ Puede ser necesario complementarlo con otros procesos ágiles como XP.
- ✓ Un mal uso de la metodología puede dar lugar a un desarrollo sin final.
- ✓ Si no se tiene experiencia en seguir procesos de desarrollo, puede ser caótica su uso.
- ✓ Si no existe una fecha definitiva de finalización del proyecto es posible que se siga solicitando, y añadiendo, nueva funcionalidad.

2.1.5.4. Prácticas

A continuación se enumeran algunas de las prácticas de Scrum [7]:

- Los clientes se convierten en parte del equipo de desarrollo.
- Scrum tiene frecuentes entregables intermedios con funcionalidad que funciona, como otras formas de procesos de software ágiles. Esto permite al cliente conseguir trabajar con el software antes y permite al proyecto cambiar los requisitos de acuerdo con las necesidades.
- Se desarrollan planes de riesgos y mitigación frecuentes por parte del equipo de desarrollo, la mitigación de riesgos, la monitorización y la gestión de riesgos se lleva a cabo en todas las etapas y con compromiso.

⁷ CLINTON KEITH, *Agile Game Development With Scrum*, 2010

- Transparencia en la planificación y desarrollo de módulos, permitir a cada uno saber quién es responsable de qué y cuándo.
- Frecuentes reuniones de las personas involucradas en el negocio para monitorizar el progreso.
- Nadie es penalizado por reconocer o describir un problema imprevisto.

2.1.5.5. Valores

Scrum es una metodología muy simple en su composición, sin embargo sus fundamentos teóricos y los valores en los que se fundamenta es lo que hace a su complejidad [⁸].

- **Compromiso:** Estar dispuesto para comprometerse a una meta. La metodología da a las personas la autoridad que necesitan para cumplir con sus compromisos.
- **Enfoque, haz tu trabajo:** Enfoca todos tus esfuerzos y habilidades para trabajar en lo que te comprometiste a hacer. No te preocupes por nada más. Alguien lo hará por ti.
- **Auto-organización:** Los equipos son auto-organizados con márgenes de decisión suficientes para tomar las decisiones que se consideren oportunas en los sucesivos Sprints.
- **Apertura / honestidad:** SCRUM mantiene todo acerca del proyecto visible a todos.
- **Respeto:** Los individuos estamos formados por nuestros orígenes y nuestras experiencias. Es importante respetar las diferentes a las personas del equipo y sus formas de pensar.
- **Coraje:** Tener el coraje para comprometerse, actuar, ser honesto y esperar respeto.

⁸ KEN SCHEWABER, JEFF SUTHERLAND, *La Guía Definitiva de Scrum*, 2011

2.1.5.6. Roles

No existe el rol de gerente de proyecto en Scrum, las responsabilidades del clásico gerente de proyecto se encuentran divididas en los tres roles principales que conforman el Equipo Scrum: el Product Owner gestiona el producto y el retorno de la inversión, el ScrumMaster gestiona el proceso y el equipo se gestiona a sí mismo [⁹].

a) Roles Principales

La metodología define tres roles principales entre los cuales se dividen todas las responsabilidades de un proyecto.

Product Owner

Está a cargo del proyecto y cerciora de que el equipo Scrum trabaja de forma adecuada. Es quien maneja y prioriza las funcionalidades a desarrollar y las coloca en el Product Backlog.

Scrum Master

Es el responsable de que los miembros del equipo sigan el proceso como es debido y de remover los obstáculos que impiden que el equipo alcance el objetivo del sprint. El Scrum Master no es el líder del equipo (porque cada miembro del equipo se auto-organiza), sino actúa como una protección entre el equipo y cualquier influencia que le distraiga.

Scrum Team

Es un equipo multidisciplinario y auto organizado, su cometido principal es construir el producto que el Product Owner especifica. El equipo puede ser de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo.

b) Roles Auxiliares

Los roles auxiliares en los equipos Scrum son aquellos que no tienen un rol formal y no se involucran frecuentemente en el proceso Scrum, sin embargo deben ser tomados en cuenta.

⁹ CLINTON KEITH, *Agile Game Development With Scrum*, 2010

Stakeholders (Clientes, Proveedores, Vendedores, etc)

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su producción. Sólo participan directamente durante las revisiones del sprint.

Managers

Es la gente que establece el ambiente para el desarrollo del producto.

2.1.5.7. Reuniones

2.1.5.7.1. Daily Scrum

El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad, al poner de manifiesto puntos en que se pueden ayudar unos a otros.

Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para al finalizar la reunión poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso conjunto que el equipo adquirió para la iteración (en la reunión de planificación de la iteración).

Cada miembro del equipo debe responder las siguientes preguntas en un time box de cómo máximo 15 minutos:

- ¿Qué he hecho desde la última reunión de sincronización? ¿Pude hacer todo lo que tenía planeado? ¿Cuál fue el problema?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener para cumplir mis compromisos en esta iteración y en el proyecto?

Beneficios

- Aumentar la productividad en el proyecto y potencia el compromiso de equipo, dado que cada miembro pone de manifiesto delante del resto:

- Las tareas que pueden afectar a otros miembros del equipo, por que impactan en su trabajo o porque hay dependencias (especialmente si existe un retraso).
- Los impedimentos con que se encuentra. La reunión de sincronización permite identificar más problemas a tiempo. El resto de miembros del equipo pueden ofrecer ayuda a otros en la realización de tareas o para resolver problemas que ya tuvieron anteriormente. El Facilitador (Scrum Master) se encargará de solucionar los impedimentos que el equipo no puede solucionar por sí solo o que le quitan tiempo para cumplir con su compromiso fundamental de desarrollo de requisitos.
- Las tareas no planeadas que está realizando que el equipo no conoce y puede que no estén alineadas con el compromiso del equipo, aunque él crea que lo que está haciendo es lo mejor que se puede hacer.
- Cuáles son sus necesidades. Cada miembro entiende las necesidades de los otros miembros del equipo respecto a su trabajo, de manera que pueden colaborar y adaptar sus trabajos para que den el máximo valor y no realizar tareas que no proporcionan ningún beneficio al resto del equipo.
- Cuál es su ritmo de trabajo. Se hace visible si de manera continua un miembro del equipo está realizando tareas por debajo del rendimiento esperado.
- Cuáles son los criterios que está utilizando para realizar sus tareas, de manera que estén alineados con los objetivos comunes del equipo.
- Fomentar el aprendizaje de los miembros del equipo, debido a que pueden ver cómo trabajan los otros según sus especialidades y experiencias.
- Conocer el estado de la iteración, ver si es posible completar los requisitos a que se comprometió el equipo, en vista de las desviaciones y de las tareas pendientes.

Restricciones

- La reunión diaria de estado y sincronización del equipo no es para resolver problemas, los problemas se resuelven después de la reunión.

- No a todos los miembros del equipo les interesan todos los detalles de cada tema.
- En la reunión los miembros del equipo programan reuniones entre ellos donde colaborar sincronizando tareas, ayudando a resolver problemas, etc.
- No puede haber una persona explicando su estado mientras otras se han apartado de la reunión para comentar un tema particular.
- El equipo debe contar con unos criterios consensuados sobre el proceso de ejecución de las de tareas
 - El proceso de ejecución de las tareas debe estar consensuado para evitar que cada reunión sea una exposición de discrepancias entre los miembros del equipo.

2.1.5.7.2. Scrum de Scrum

Esta reunión se realiza cada día normalmente después del Daily Scrum:

- Estas reuniones permiten a los grupos de equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración.
- Asiste una persona asignada por cada equipo.

La agenda será la misma que la del Daily Scrum, añadiendo además las siguientes cuatro preguntas:

- ¿Qué ha hecho tu equipo desde nuestra última reunión?
- ¿Qué hará tu equipo antes que nos volvamos a reunir?
- ¿Hay algo que demora o estorba a tu equipo?
- ¿Estás a punto de poner algo en el camino del otro equipo?

2.1.5.7.3. Reunión de Planificación del Sprint (Sprint Planning Meeting)

La reunión de planificación de Sprint es atendido por el Product Owner, Scrum Master, todo el Equipo Scrum, y representantes de la dirección o cualquier cliente interesado y apropiado. Durante la reunión el Product Owner describe las funciones de más alta prioridad para el equipo. El equipo hace preguntas suficientes durante esta reunión para

que puedan apagarse después de la reunión y determinar las tareas que van a pasar de la acumulación de productos para el Sprint Backlog. Colectivamente, el equipo de Scrum y el Product Owner definen la meta del Sprint, que es una breve descripción de lo que el sprint intentará lograr. Después de la reunión de planificación de Sprint, el equipo Scrum se reúne por separado para discutir lo que escucharon y decidir que se pueden comprometer a realizar durante el sprint. En algunos casos habrá que negociar con el Product Owner, pero siempre estará con el equipo para determinar la cantidad que puede comprometerse a completar.

Al inicio del ciclo Sprint (cada 15 o 30 días), en una Reunión de Planificación del Sprint se lleva a cabo lo siguiente:

- Seleccionar qué trabajo se hará
- Preparar, con el equipo completo, el Sprint Backlog que detalla el tiempo que tomará hacer el trabajo.
- Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint
- Ocho horas como límite

La reunión de Planificación del Sprint consiste de dos partes, cada una de las cuales puede durar hasta 4 horas.

- 1) Selección del backlog (Exposición):** El Product Owner presenta el Backlog de mayor prioridad al equipo de desarrollo. Juntos colaboran para acordar cuántos de estos items pueden convertirse en un Incremento del Producto potencialmente productivo durante el próximo Sprint. El equipo realiza la Estimación del Sprint y selecciona todo lo que se pueda realizar.
- 2) Planificación de trabajo del Sprint (Resolución):** El equipo define la arquitectura y el diseño de la funcionalidad que fue seleccionada, y luego define el trabajo, o tareas que conformarán el Backlog del Sprint, para construir dicha funcionalidad durante el próximo Sprint.

2.1.5.7.4. Reunión de Revisión del Sprint (Sprint Review Meeting)

La Revisión del Sprint brinda una inspección del progreso del proyecto al finalizar cada Sprint. Basándose en esta inspección se realizan adaptaciones al proyecto. Al fin del Sprint el equipo presenta el incremento del Producto que pudo construir. El management, clientes, usuarios y el Product Owner verifican el incremento del Producto. Escuchan las historias del equipo que surgieron durante el transcurso del Sprint. Escuchan qué salió mal y qué salió bien. Verifican en dónde realmente se encuentran en la construcción de todo el producto. Luego de todo esto, finalmente pueden tomar una decisión con información acerca de qué hacer a continuación. En otras palabras, determinan el mejor curso de acción para poder alcanzar el destino.

Durante la reunión, todos ven funcionar la demostración del producto en el entorno del cliente o usuario. A medida que se visualiza, consideran qué funcionalidad podría ser agregada durante el próximo Sprint. El incremento del producto es el punto inicial para una lluvia de ideas.

A medida que el equipo muestra el incremento del producto, ayuda a los espectadores a entender las fortalezas y debilidades del producto, y las dificultades y éxitos de la experiencia de trabajar juntos.

2.1.5.7.5. Retrospectiva Del Sprint (Sprint Retrospective)

En Scrum, la Retrospectiva del Sprint es una reunión facilitada por el Scrum Master en la cual los Miembros del Equipo de Scrum discuten el Sprint que acaba de finalizar, y determinan qué podría cambiarse en el próximo Sprint para que sea más productivo y mejor. La Revisión del Sprint se focaliza en "Qué" construye el equipo, mientras que la Retrospectiva se centra en "Cómo" están construyendo el producto.

Cualquier cosa que afecte al equipo la construcción del producto está abierta para discutirse. Esto puede incluir: procesos, prácticas, comunicación, entorno, artefactos y herramientas.

Scrum no es una metodología prescriptiva, sino más bien un framework que debe adaptarse apropiadamente para cada proyecto, equipo y circunstancias particulares. La

Retrospectiva del Sprint es un mecanismo importante que le permite al equipo evolucionar continuamente y mejorar durante toda la vida del proyecto.

2.1.5.8. Sprint

Un Sprint en Scrum es el término que denomina a una iteración que está acotada en el tiempo, usualmente entre 2 y 4 semanas, durante la cual el equipo trabaja para convertir items del Backlog del Producto en un incremento del Producto potencialmente productivo [¹⁰].

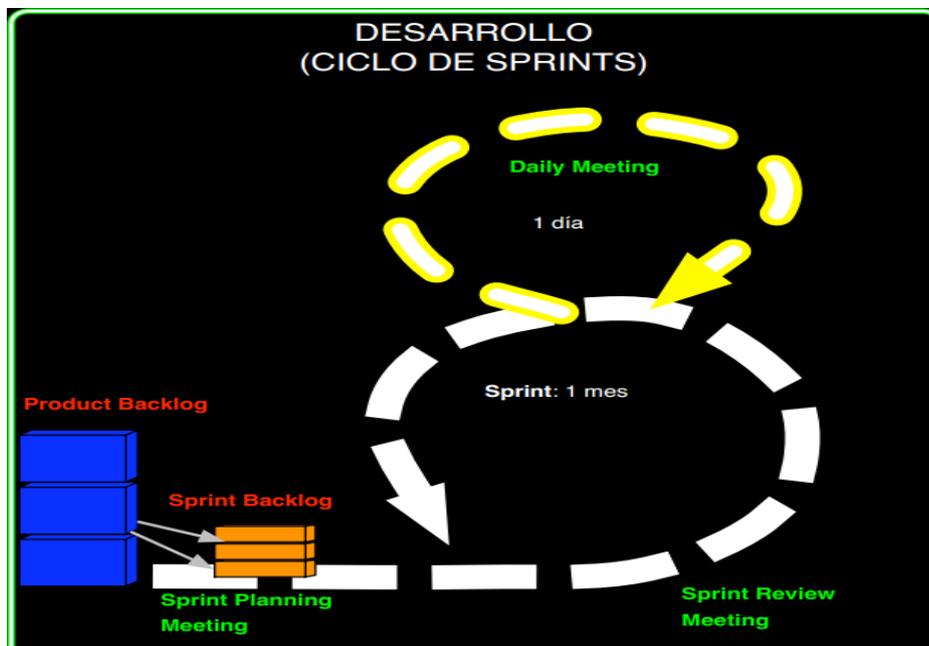


Figura II.3. Ciclo de Sprints de Scrum

Fuente: <http://jorgemanrubia.net/blog/wp-content/uploads/2009/06/2009-06-CharlaPreparaticAgil.pdf>

La duración del Sprint debería ser lo suficientemente larga para crear algo de valor y con la suficiente calidad como para ser demostrado frente al Product Owner y los stakeholders. Con un plazo superior a 4 semanas el Equipo pierde agilidad, debido a que tendrá más artefactos y documentación por la cual preocuparse.

Un Sprint de 2 semanas es la duración ideal y se convirtió en un estándar defecto por los siguientes motivos:

¹⁰ CLINTON KEITH, *Agile Game Development With Scrum*, 2010

- Fuerza al equipo a operar de una manera ágil, en vez de hacer "mini cascadas".
- Brinda un feedback más frecuente en lo que se está construyendo.
- Reduce el riesgo de "construir la cosa equivocada".
- Incrementa la respuesta del equipo ante cambios en el negocio.
- Le brinda al equipo más oportunidades de analizar y adaptarse a la forma que trabajan.

Los objetivos y actividades del Sprint se planifican al comienzo de cada Sprint en la reunión de Planificación del Sprint. Al finalizar el Sprint el equipo demuestra lo que construyeron en la Revisión del Sprint, y analizan su propia performance y deciden qué mejorar en la Retrospectiva del Sprint.

La duración de las reuniones del Sprint (Planificación, Revisión y Retrospectiva) se escalan de acuerdo a la duración del Sprint.

El progreso del Sprint se sigue con el Backlog del Sprint y un gráfico Burn-Down.

Un Sprint consta de las siguientes actividades:

- Elaborar
- Integrar
- Revisar
- Ajustar.

2.1.5.9. Fases

Scrum se estructura en tres fases denominadas pre-game, game y post-game [¹¹].

¹¹ CLINTON KEITH, *Agile Game Development With Scrum*, 2010

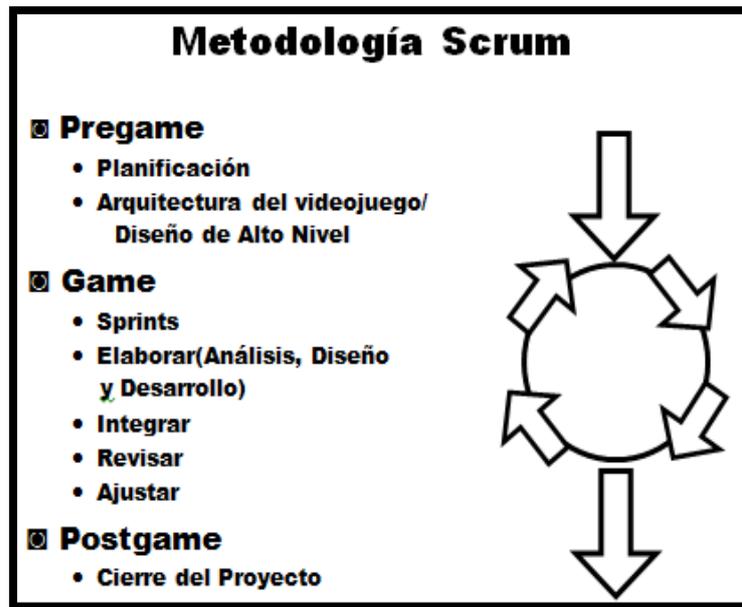


Figura II.4. Fases de Scrum

Fuente: <http://www.scrummanager.net/blog/author/jpalacio/>

Pre-game

En esta fase se define el producto basado en las características conocidas, estimando su tiempo y costo. También se analiza el sistema a construir, se define la arquitectura y se realiza un diseño de alto nivel de la solución.

El juego debe ser descubierto, es decir, lo que realmente son los objetivos del juego y lo que sería el factor diversión. Se escoge el lenguaje de programación, plataforma y otras herramientas necesarias para el desarrollo. Pre-game reduce la necesidad de encontrar ese elemento elusivo de diversión durante la etapa de Game, y permite al equipo centrarse en la ejecución del juego, en lugar de experimentar con él.

Define el sentido que tendrá la fase de Game, sin ahogar la creatividad que pueda surgir allí. Aquí el trabajo será encontrar el concepto del juego ideal y diseño.

También el desarrollo de un prototipo ayudará a visualizar el factor diversión que el juego o una parte de él puede ofrecer. El prototipo debe proporcionar una navegación básica y simplificada para el usuario y las funciones necesarias para poner a prueba. Por lo general, el código producido no se utiliza de nuevo, y el prototipo se descarta.

Pasos de la planificación:

- * Desarrollo de un backlog completo.
- * Determinación de la fecha de entrega y la funcionalidad de una o más versiones.
- * Selección de la versión más adecuada para desarrollo inmediato.
- * Trazado de los paquetes del producto (objetos) sobre los elementos del backlog de la versión elegida.
- * Selección del equipo o equipos para desarrollar la nueva versión.
- * Evaluación y control adecuado de los riesgos.
- * Estimación del coste de la versión, incluyendo desarrollo, material, marketing, formación y despliegue.
- * Conformidad de la dirección y financiación del proyecto.

Pasos de la arquitectura:

- * Revisión de los elementos del backlog incluidos en la versión.
- * Identificación de los cambios necesarios para implementar el backlog.
- * Análisis del dominio para incluir los requisitos que incluye el desarrollo mejorado y actualización.
- * Acotar la arquitectura del sistema para apoyar el nuevo contexto y necesidades.
- * Identificar problemas del desarrollo o modificaciones.
- * Reunión de revisión de diseño. Cada equipo presenta los cambios para implementar los elementos del backlog, e identificar posibles reasignaciones

Game

En esta etapa, se tiene ya el alcance del proyecto bien definido, y por lo tanto una buena idea de lo que realmente es el juego y lo que se debe hacer.

La fase de Game que suele llamarse Sprint o fase de desarrollo, consta de iteraciones que pueden durar de 1 a 4 semanas (por lo general 1-2 semanas), donde se desarrollan las funcionalidades del juego, su duración debe ser constante en cada ciclo. Sin embargo, este requisito no es estricto debido a que a veces el tiempo de cada sprint es diferente.

Al comienzo de cada iteración se realiza su planificación, donde se describen, priorizan y estiman las funcionalidades que se van a desarrollar y al concluir se evalúa su resultado.

En el comienzo de cada sprint en la reunión de planificación se realiza lo siguiente:

- Análisis del Backlog (Máximo 4 horas).
 - Se traduce las entradas del Backlog a tareas reales y demostrables.
 - Jugar al póker para determinar la complejidad usando una escala del 1, 2, 3, 5, 8, 13, 21 (es una gran herramienta para comunicar entre todos los miembros del equipo la complejidad del producto).
 - El Product Owner debe dar su visto bueno.

- Planificación de tareas (máximo 4 horas)
 - Se descompone cada tarea en subtareas.
 - Se decide quien trabaja en que.
 - Se minimizan dependencias.
 - Se estiman las horas de trabajo.

¡Scrum ocupa 5 horas al día! (Las otras 3 se ocupan para mitin, comunicación, imprevistos o consultas que pueden salir)

- Se realiza el Standup por la mañana
 - Máximo 3 minutos por miembro.
 - ¿Qué hizo ayer?
 - ¿Qué hará hoy?
 - ¿Algún problema?

- El resto del día el miembro del Scrum reparte sus tareas como mejor le convenga.
 - ¡Tener cuidado con las dependencias!

Y al final de cada iteración se lleva a cabo una reunión de revisión, en donde se presenta la nueva funcionalidad del producto, las metas incluyendo la información de las funciones, diseño, ventajas, inconvenientes y esfuerzos del equipo, es decir:

- Se muestra al Product Owner el resultado del Sprint.
- Cada Scrum master organiza la presentación.
- Cada miembro del Scrum muestra sus resultados.
- Al finalizar la revisión se realiza el postmortem
 - Cada Scrum se reúne analizar el sprint.
 - ¿Qué fue bien?
 - ¿Qué puede mejorar?

Post-game

Esta fase comienza cuando el equipo de management decide que las variables de entorno, tales como los requerimientos se han completado. Aquí se realiza el cierre del proyecto, donde se prepara la liberación del videojuego, se verifican las versiones a entregar, se genera la documentación final y se realiza el pre-lanzamiento y el lanzamiento.

2.1.5.10. Artefactos

Scrum, propone cinco artefactos para mantener organizados los proyectos. Estos artefactos, ayudan a planificar y revisar cada uno de los Sprints, aportando medios ineludibles para efectuar cada una de las reuniones.

2.1.5.10.1. User Story

Cada user story contiene lo que el cliente del juego quiere, con total claridad y ninguna ambigüedad. Deben ser breves, concretas y algo estimables. Son el resultado de escuchar al cliente y ayudarlo a resumir el requisito en una sola frase. Algo muy importante es que están escritas con el vocabulario del cliente, no con vocabulario técnico como se muestra en la **Figura II.5.:**

Como un / una	Quiero.....	De modo que.....
Asesor	Crear un nuevo juego introduciendo un nombre y una descripción opcional	Puedo empezar invitando estimadores
Asesor	Invitar estimadores dándoles una url donde se puede acceder al juego	Podemos iniciar el juego
Estimador	Unirse a un juego por entrar en mi nombre en la página que recibió la url para	Puedo participar
Asesor	Comienza una nueva ronda mediante la introducción de un elemento en un solo campo de texto de varias líneas	Podemos estimar
Animador	Cambiar animaciones directamente en el juego sin reiniciarlo para que	Pueda iterar más rápido en las animaciones

Figura II.5. User Story de Scrum

Fuente: Los Autores.

Cada user story provoca una serie de preguntas acerca de los múltiples contextos en que se puede dar. Son las que naturalmente hacen los desarrolladores a los analistas de negocio o al cliente, por ejemplo:

- ♣ ¿Qué hace el videojuego si el arma que se quiere añadir al personaje ya está dentro de él?
- ♣ ¿Qué sucede si se ha agotado las balas del arma?
- ♣ ¿Se le indica al usuario que el arma ha sido añadido al personaje?

Las respuestas a estas preguntas son afirmaciones, ejemplos, los cuales se deben transformar en test de aceptación.

2.1.5.10.2. Product Backlog

Es un documento de alto nivel para todo el proyecto. Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc., como se muestra en la **Figura II.6.**, priorizadas según su retorno sobre la inversión (ROI). Es el qué va a ser construido. Es abierto y cualquiera puede modificarlo. Contiene estimaciones grosso modo, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido. Esta estimación ayuda al Product Owner a ajustar la línea temporal y, de manera limitada, la prioridad de las diferentes tareas. Por ejemplo, si dos características tienen el mismo valor de negocio la que requiera menor tiempo de desarrollo tendrá probablemente más prioridad, debido a que su ROI será más alto.

Prioridad	Item	Estimación	Estado	Criterios de aceptación
1	Como asesor quiero crear un nuevo juego introduciendo un nombre y una descripción opcional para poder empezar invitando estimadores	3	En curso	# Se debe ingresar los nombres completos. # La descripción no debe ser mayor de 50 caracteres.
2	Como animador quiero cambiar animaciones directamente en el juego sin reiniciarlo para que pueda iterar más rápido en las animaciones	5	Pendiente	# Cuando se cambie las animaciones se debe guardar el estado del juego.

Figura II.6. Product Backlog

Fuente: Los Autores.

2.1.5.10.3. Sprint Backlog

Es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas con ninguna tarea de duración superior a 16 horas (**Figura II.7.**). Si una tarea es mayor a 16 horas, deberá ser dividida en otras menores. Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

Requisito	Tarea	Quien	Estado	Día								
				1	2	3	4	5	6	7	8	
				Horas	120	80	76	40	82	80	92	72
				pendientes								
Requisito A	Tarea 1	Gabriel	Completada		16	8						
Requisito A	Tarea 4	Milton	Completada		4							
Requisito A	Tarea 5	Milton	Completada		4							
Requisito A	Tarea 3	Gabriel	Completada		24	16	4					
Requisito B	Tarea 2	Gabriel	En progreso		18	8	8	8	16	16	16	8
Requisito B	Tarea 6	Milton	En progreso		18	15	8	16	8	14	14	10
Requisito C	Tarea 7	Milton	No iniciada		8	16	16	8	16	16	8	8
Requisito C	Tarea 8	Gabriel	No iniciada		8	24	16	8	8	16	8	16

Figura II.7. Sprint Backlog

Fuente: Los Autores.

2.1.5.10.4. Burn-Down

Es un gráfico que muestra cuánto le falta al equipo para completar con el compromiso del Sprint. Sobre el eje horizontal se ubican los días hábiles que tiene el Sprint, y sobre el eje vertical la cantidad de puntos que tienen que quemar durante ese Sprint (**ver Figura II.8.**).

La cantidad de puntos a quemar se calcula durante la Planificación del Sprint, en donde el equipo se compromete ante el Product Owner a realizar cierta cantidad de tareas. La suma de las estimaciones de estas tareas es la cantidad de puntos que el equipo deberá quemar durante el Sprint para cumplir con el compromiso.

El gráfico se actualiza diariamente, durante la Reunión Diaria de Scrum, el equipo actualiza las estimaciones de las tareas, y así va contando cuántos puntos le quedan por terminar en el Sprint.

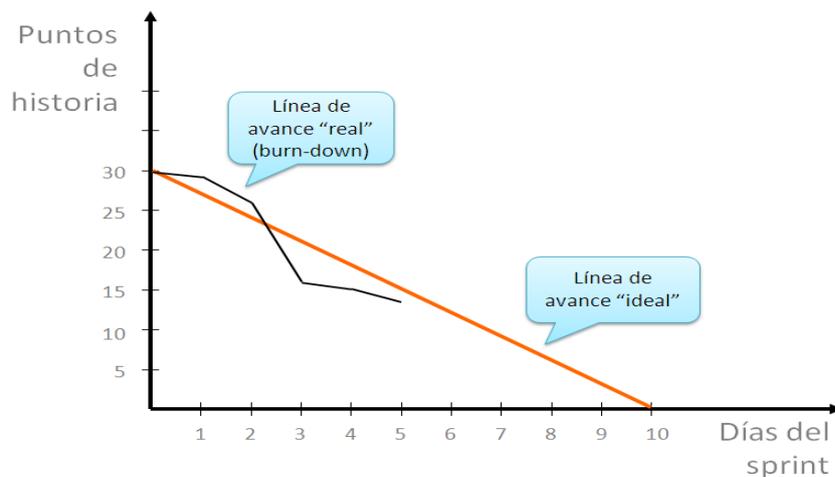


Figura II.8. Gráfico de Burn-Down

Fuente: http://www.dosideas.com/wiki/Archivo:Burn-down_de_scrum.png/

2.1.5.10.5. Burn-Up

Es un gráfico que muestra cuánto avanzó el equipo hasta el momento, cuántos puntos llegaron a quemar hasta el último Sprint, en relación al esfuerzo total necesario para terminar el proyecto.

Sobre el eje horizontal se ubican los Sprints, sobre el eje vertical se ubican los puntos de Stories (ver **Figura II.9.**).

Sabiendo el factor de foco del equipo y su cantidad de integrantes podemos saber su velocidad (cuántos puntos quemarán en cada sprint). Con estos datos se puede trazar la "línea de quemado ideal", la cual indica el avance ideal para llegar a quemar todos los puntos del proyecto. De aquí se deriva la cantidad de sprints que necesitará el equipo para terminar el proyecto.

Al finalizar cada Sprint el equipo sabe cuántos puntos pudo quemar en esa iteración, y va acumulando este valor en el gráfico.

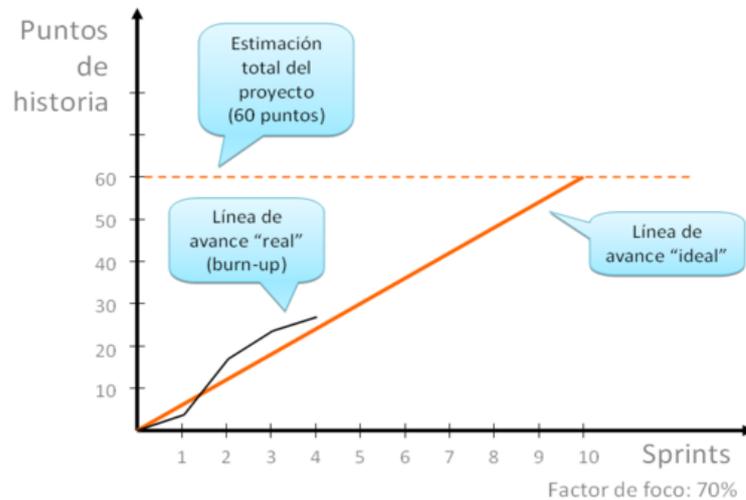


Figura II.9. Gráfico de Burn-Up

Fuente: http://www.dosideas.com/wiki/Archivo:Burn-down_de_scrum.png/

2.1.6. Metodología Extreme Game Development

Fue descrita en 2003 por Thomas Demachy a través de un documento presentado al portal Gamasutra.

XGD se define como una metodología ágil para la producción de videojuegos, es una adaptación de eXtreme Programming, una práctica muy común en el desarrollo de software ágil.

El XGD fue desarrollado por la empresa Titus Interactive Studio para la que trabajó Demachy, que luego desarrolló videojuegos.

Según Demachy, muchos equipos de desarrollo de videojuegos han utilizado el Extreme Programming como metodología estándar. Sin embargo, XP tiene una limitación importante cuando se aplica al universo de los videojuegos: su creación principalmente fue para satisfacer las necesidades de los programadores. Desde un proyecto de Juego electrónico involucra a personas de diversas disciplinas, por ello vieron la necesidad de la adaptación de XP por lo que podría absorber esta peculiaridad de los proyectos de videojuego, la electrónica. Como resultado de esta adaptación, se produjo el Extreme Game Development.

XGD permite a los editores cambiar de opinión durante el proceso de desarrollo, evolucionar a la tecnología, a los desarrolladores estar orgullosos de sus logros y completar su juego a tiempo.

XGD hace hincapié en una estrecha colaboración entre el editor y desarrollador, en lugar de depender de un contrato basado en la negociación. Los contratos son esenciales en la relación creador-editor, pero ambas partes tienen que trabajar juntos para crear un juego de calidad al final del tiempo de desarrollo.

Por otro lado, el desarrollador debe estar de acuerdo para ofrecer una versión jugable del juego tan pronto como sea posible, que se actualice en cada hito. De esta forma el editor puede decidir si hay contenido suficiente en el juego para liberarlo [¹²].

2.1.6.1. Características

Estas son algunas de las características de XGD:

- El editor debe estar presente en el equipo de trabajo de desarrollo.
- Etapas cortas de seis semanas, donde se realiza la planificación y la elección de las funciones más importantes a ser desarrollados primero.
- Ciclos de desarrollo de tres semanas, donde las tareas están definidas, estimadas y desarrolladas.
- Integración continua soportada por herramientas apropiadas.
- Todos son parte del equipo, por lo que cada uno es responsable del videojuego.
- Pruebas automatizadas.
- Versiones.
- Uso del Lenguaje de Modelado Unificado (UML).

2.1.6.2. Ventajas

- ✓ Permite tener una orientación para aumentar al cliente la posibilidad de que el juego producido realmente satisfaga sus necesidades.

¹² http://www.gamasutra.com/resource_guide/20030714/demachy_pfv.htm

- ✓ Disminuye el riesgo del proyecto al demostrar que su método funciona y poniendo funcionalidad en las manos de los clientes, lo que les permite reaccionar a tiempo en el desarrollo del videojuego.
- ✓ Permite realizar pruebas de integración continua y ayuda a aumentar la calidad del videojuego.
- ✓ Es atractivo para los programadores, diseñadores músicos, guionistas, productores y artistas que normalmente no están dispuestos a adoptar un proceso en el desarrollo del videojuego, permitiendo a su grupo de trabajo gestionar actividades con el fin de mejorar el juego.
- ✓ Permite realizar una programación organizada del videojuego.
- ✓ Tiene una menor tasa de errores en el desarrollo del videojuego.
- ✓ Existe satisfacción del equipo al momento de desarrollar el videojuego.
- ✓ Solicita a los desarrolladores mejorar de forma dinámica el juego en su conjunto.
- ✓ Aumenta el conocimiento del desarrollador del videojuego.

2.1.6.3. Desventajas

- ✓ Está dirigido a un solo videojuego, desarrollado y mantenido por un solo equipo.
- ✓ Es particularmente vulnerable a los desarrolladores que no funcionan bien con los demás, que piensan que lo saben todo, o que no están dispuestos a compartir su código del videojuego.
- ✓ No va a funcionar en un entorno en el que un cliente insiste en una especificación completa o diseño del videojuego antes de comenzar la programación.
- ✓ No va a funcionar en un entorno donde los programadores, diseñadores músicos, guionistas, productores y artistas están separados geográficamente.
- ✓ Es recomendable emplearlo solo en videojuegos cuyo desarrollo sea a corto plazo.
- ✓ Altas comisiones en caso de fallar el desarrollo del videojuego.
- ✓ La refactorización no siempre puede ser apropiada al momento de desarrollar el videojuego

2.1.6.4. Prácticas

XGD mantiene algunas de estas prácticas, con poco o ningún cambio en relación a su base, la Extreme Programming. A continuación se muestra algunas de estas prácticas que son sostenidas [¹³]:

- **Whole Team (Equipo Completo):** El equipo debe permanecer en comunicación coherente. El equipo es un todo, no lo suma las fuerzas individuales.
- **Incremental Design (Diseño incremental):** Las tareas (activos) de un juego deben ser de forma más sencilla posible, que funcionen correctamente.
- **User Stories (Historias de usuario):** Se trata de una breve descripción de las funcionalidades del juego, por lo general descrito por el cliente. A menudo, tienen el documento los Publishers.
- **Weekly Cycle (Ciclo Semanal):** El proyecto está organizado y planificado para ser ejecutado en ciclos de corta duración.
- **Continuous Integration (Integración continua):** En el proyecto existen, por ejemplo, trozos de código fuente que continuamente se integran en funcionamiento.
- **Shared Code (Código compartido):** Todo el equipo compartirá el código fuente del proyecto.
- **Stand-up Meetings (Reuniones diarias de seguimiento):** Reuniones rápidas del proyecto, con el objetivo de que todo el equipo esté consciente de la labor que se está realizando en un momento dado. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar. Para no quitar tiempo innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie.

¹³ http://di.uern.br/sebastiao/wp-content/uploads/2010/04/raphael_monografia.pdf
<http://c2.com/cgi/wiki?ExtremeProgrammingForGames>

- **The planning game (El juego de la planificación):** Hay una comunicación frecuente entre el cliente y el equipo de desarrollo del videojuego. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- **Deliveries small (Entregas pequeñas):** Producir rápidamente versiones del videojuego que sean jugables, aunque no cuenten con toda la funcionalidad del videojuego. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más de 3 meses.
- **Testing (Pruebas):** La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del videojuego.
- **Refactoring (Refactorización):** Es una actividad constante de reestructuración del código con el objetivo de evitar duplicación del código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- **Development in pairs (Desarrollo en parejas):** Toda la producción del videojuego debe realizarse con trabajo en parejas pero simplemente los programadores y diseñadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción del equipo).

2.1.6.5. Valores

XGD hace hincapié los cuatro valores fundamentales de XP: simplicidad, comunicación, retroalimentación y valentía [¹⁴].

Simplicidad: Lo más importante es la simplicidad, no siempre se necesita que el juego sea innovador en todos los géneros, debe tener lo necesario.

¹⁴ http://www.gamasutra.com/resource_guide/20030714/demachy_pfv.htm

Comunicación: Se debe priorizar la comunicación de forma oral, de manera que las personas se sientan más cómodas para expresarse, no necesariamente con una gran cantidad de documentos cuya utilidad puede verse reducida (en número de personas que la utilicen), sin olvidar que existen documentos indispensables tales como: las propuestas del juego, los conceptos del juego, el documento del diseño del juego, entre otras.

Retroalimentación: Está fuertemente ligada a la comunicación. La mejor manera de conocer el estado actual del videojuego es haciéndole pruebas funcionales, esto proporcionará información real y confiable sobre el grado de fiabilidad.

Coraje: A primera vista esto puede parecer un valor extraño para un método de desarrollo, pero el valor está involucrado en cada paso del proceso. Se necesita coraje para que se empiece a crear el videojuego, en ocasiones, sin que todo esté determinado, manteniendo incertidumbre por algún tiempo.

2.1.6.6. Roles

Existen diferentes roles y responsabilidades en XGD para diferentes tareas y propósitos durante el proceso, clasificándose en:

a) Roles principales

Development Team (Equipo de desarrollo)

Es un equipo interdisciplinario como diseñadores, programadores, artistas, músicos, guionistas, entre otros, son responsables de: construir el videojuego, tomar decisiones técnicas y realizar las pruebas.

Manager (Jefe de Proyecto)

Es el responsable de organizar y guiar las reuniones, además asegura las condiciones adecuadas para el proyecto.

Customer (Cliente)

Es parte del equipo, escribe las historias de usuario y las pruebas funcionales para validar la implementación del videojuego. Además, asigna la prioridad a las historias

de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas en el desarrollo del videojuego.

Tester (Encargado de Pruebas)

Ayuda al cliente a escribir las pruebas funcionales del videojuego. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para las pruebas.

Tracker (Rastreador)

Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

Coach (Entrenador)

Es el responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XGD y se siga el proceso correctamente.

Gestor (Big boss)

Es el vínculo entre clientes y desarrolladores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

b) Roles opcionales

Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el desarrollo del videojuego. Guía al equipo para resolver un problema específico.

Doomsayer (Augur de desastres)

Es el responsable de que se conozcan los riesgos involucrados, que las malas noticias sean notificadas en la medida correcta, y que no se sobredimensionen. Busca posibles riesgos en forma constante, presentándolos al equipo siempre con una posible solución.

2.1.6.7. Reuniones

XGD mantiene reuniones de: Planning Game, Iteration Meeting y Daily Stand Up Meeting

2.1.6.7.1. Planning game

Es una reunión entre todos los actores del proyecto (cliente, desarrolladores, directores, etc.), para fijar el cronograma de entregas donde se establece qué user stories serán agrupadas para conformar una entrega, típicamente el cliente ordenará y agrupará según sus prioridades los user stories.

El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores.

Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

2.1.6.7.2. Iteration Meeting

Se realiza una reunión de planificación en detalle al comienzo de cada iteración, lo que se hace es:

- Decidir cuáles son los user stories que hay que implementar en la iteración. Esta decisión le corresponde en su mayor parte al usuario.
- Escribir los tests de aceptación que tendrán que ser satisfechos por cada user story.

- Los user stories y los tests se traducen en tareas, que se escriben en tarjetas. Cada una debería durar entre uno y tres días. El plan detallado para cada iteración consiste en ese conjunto de tareas.
- Al igual que ocurre con los user stories, las tareas más cortas de un día se agrupan con otras, y las más largas de tres días se dividen en otras más pequeñas.
- Las tareas se asignan a programadores, diseñadores y artistas concretos y son estos los que estiman el tiempo que puede tardar la implementación de la tarea que se compromete a hacer.
- Si la iteración va retrasada con respecto a lo proyectado se escogen algunos user stories y se dejan para la siguiente iteración. Igualmente, si va adelantado se escogen user stories y se introducen en la iteración actual.
- La velocidad a la que va el proyecto se estima en función del tiempo que han tardado en implementarse los user stories que ya están hechos. Cada tres a cinco iteraciones habrá que reestimar el tiempo de los stories que quedan.
- Es importante que no se planifique en detalle más que lo que se va a hacer en la iteración en curso. Motivo: Es perder el tiempo, el proyecto cambiará.
- No se deben añadir funcionalidades extras antes de tiempo, es decir, no implementar funcionalidades que no son necesarias en el momento. Motivo: El 90% no serán usadas.

2.1.6.7.3. Daily Stand Up Meeting

El propósito de la reunión stand up es la comunicación entre el equipo de desarrollo, todas las mañanas se usa para comunicar problemas, soluciones, y promover el enfoque de equipo, realizando largas discusiones entre sus miembros. Durante la reunión los desarrolladores del videojuego informan por lo menos tres cosas: lo que se llevó a cabo ayer, lo que se intentará hoy, y qué problemas están causando retrasos. La reunión ayuda mucho en el desarrollo del videojuego evitando errores y retrasos por parte de los desarrolladores.

2.1.6.8. Proceso

El desarrollo es la pieza clave de todo el proceso. Todas las tareas tienen como objetivo realizar el desarrollo a la máxima velocidad, sin interrupciones y siempre en la dirección correcta. A grandes rasgos, el ciclo de desarrollo se puede simplificar en los siguientes pasos [¹⁵]:

1. El cliente define el valor de negocio a implementar.
2. Los desarrolladores estiman el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. Los desarrolladores construyen ese valor de negocio.
5. Vuelve al paso 1.

2.1.6.9. Actividades

Escuchar

Si se va a hacer pruebas se tiene que preguntar si lo obtenido es lo deseado, y preguntar a quién necesita el videojuego. Se tiene que escuchar a los clientes cuáles son los resultados deseados, tener una escucha activa explicando lo que es fácil y difícil de obtener, la realimentación entre cliente y equipo de desarrollo ayudará a entender los problemas.

Diseñar

El diseño crea una estructura que organiza al desarrollo del videojuego, un buen diseño permite que el videojuego crezca de manera sencilla, si alguna parte del videojuego es de desarrollo complejo, lo apropiado es dividirla en varias sub-tareas. En caso de haber fallas en el diseño estas deben ser corregidas cuanto antes porque de lo contrario se verán plasmadas en el videojuego disminuyendo su calidad y en ocasiones no cumpliendo con los requerimientos.

¹⁵ THOMAS DEMACHY, *Extreme Game Development: Right on Time*, 2003

Codificar

El proceso de codificación se basa en plasmar las ideas y funcionalidades del videojuego a través del código. Se recomienda que el código sea sencillo y legible para todos los integrantes del equipo.

Hacer pruebas

Las pruebas brindan la oportunidad de saber si lo implementado es lo que en verdad se tenía en mente por parte del cliente, cuando no es posible pensar en ninguna prueba que pueda originar una falla en el videojuego, entonces se da por finalizado el proceso de pruebas. Lo importante es la objetividad de las pruebas y que sean realizadas desde el punto de vista del jugador.

2.1.6.10. Fases

El ciclo de vida ideal de XGD consiste en seis fases:



Figura II.10. Fases de un proyecto en XGD.

Fuente: <http://oness.sourceforge.net/proyecto/html/ch05s02.html>

Exploración

Esta fase permite a los clientes plantear a grandes rasgos los user story que son de interés para la primera entrega del videojuego. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el desarrollo de videojuego, además se explora las posibilidades de diseño mediante una lluvia de ideas construyendo una idea y pensando en el futuro del videojuego. Esta fase

toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los desarrolladores con la tecnología.

Planificación

En esta fase el cliente establece la prioridad de cada user story sabiendo cuales características del videojuego se podrán realizar de manera rápida, los desarrolladores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma tomando acuerdos sobre el contenido de la primera entrega conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses.

Las estimaciones de esfuerzo asociadas a la implementación de los user stories establecen los desarrolladores utilizando como medida el punto de partida de imaginación para el desarrollo de las mismas, estableciendo puntos por iteración, basándose principalmente en la suma de puntos correspondientes a los user stories que fueron terminadas en la última iteración.

La planificación puede ser realizada basándose en el tiempo o el alcance del desarrollo del videojuego.

Iteraciones

Esta fase comprende de varias iteraciones sobre el videojuego antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se establece la idea entregable de todo el videojuego que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo los stories que fueren la creación del videojuego, sin embargo, esto no siempre es posible debido a que es el cliente quien decide que stories se implementarán en cada iteración en base a las prioridades. Al final de la última iteración el videojuego estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: user stories no abordadas, velocidad del videojuego, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior, aquí se detalla la honestidad de los desarrolladores es decir si están o no cumpliendo con su trabajo.

Producción

La fase de producción requiere de pruebas y comprobación extra del funcionamiento del videojuego antes de que sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas funcionalidades o características a la versión actual, debido a cambios durante esta fase, las mismas serán documentadas para su posterior implementación. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana.

Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XGD debe mantener el videojuego en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del videojuego en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Muerte del Proyecto

Esta fase se lleva a cabo cuando el cliente no tiene más funcionalidades para ser incluidas en el videojuego, satisfaciendo las necesidades del cliente en otros aspectos como rendimiento y jugabilidad del videojuego. Se genera la documentación final y no se realizan más cambios. La muerte del proyecto también ocurre cuando el videojuego no genera los resultados esperados por el cliente o cuando no hay presupuesto para mantenerlo.

2.1.6.11. Artefactos

2.1.6.11.1. Tablero XGD

Es una imagen actualizada con cada etapa ordenada, lo hace el director del proyecto mediante un análisis a partir de la información recibida hasta el momento. En ella, el gerente pone cinco prácticas y herramientas que se están utilizando, así como los cinco que les gustaría usar en la próxima etapa.

Para cada una de las prácticas o herramientas que están siendo utilizadas, hay una puntuación de cero a cinco. El puntaje histórico indica si la metodología está siendo plenamente utilizada y si el equipo está motivado para aplicarlo, o si el XGD no es la metodología correcta para el equipo de desarrollo. Así, la metodología ofrece constante reflexión y mejora los procesos, aunque centralizado en una sola persona [¹⁶].

2.1.6.11.2. User Story

Representan una breve descripción del comportamiento del videojuego, emplea terminología del cliente sin lenguaje técnico, se realiza una por cada característica principal del videojuego, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

En la **Figura II.11.**, se muestra la información que contiene un User Story.

USER STORY	
Numero: 01	Nombre: Evitar los asteroides
Usuario: Jugador	
Número de modificación de story: N/A	Iteración Asignada: Primera
Prioridad en Negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Media (Alta / Media / Baja)
Descripción: Con el cohetes se avanza y se retrocede al pulsar las flechas izquierda y derecha para poder evitar los asteroides	
Observaciones:	

Figura II.11. User Story XGD.

Fuente: Los Autores.

Los user stories tienen tres aspectos:

- **Tarjeta:** en ella se almacena suficiente información para identificar y detallar el story.
- **Conversación:** cliente y desarrolladores discuten el story para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera confirmación).

¹⁶ THOMAS DEMACHY, *Extreme Game Development: Right on Time*, 2003

- **Pruebas de Aceptación:** permite confirmar que el story ha sido implementado correctamente (ver Figura II.12.).

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nro. y Nombre):
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

Figura II.12. Caso de Prueba de Aceptación XGD.
Fuente: Los Autores.

2.1.6.11.3. Task Card

Son herramientas de los desarrolladores de planificación primaria, se usan para describir las tareas que se realizan sobre el videojuego.

Las tareas pueden ser: desarrollo, corrección, mejora, etc.

Estas tareas tienen relación con un user story; se especifica la fecha de inicio y fin de la tarea, se nombra al programador, diseñador o artista responsable de cumplirla y se describe que se tratara de hacer en la tarea.

Tarea de Ingeniería	
Número Tarea:	Historia de Usuario (Nro. y Nombre):
Nombre Tarea:	
Tipo de Tarea : Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Desarrollador Responsable:	
Descripción:	

Figura II.13. Task Card XGD.
Fuente: Los Autores.

2.1.6.11.4. Card CRC (Clase - Responsabilidad – Colaborador)

Es una técnica de diseño orientado a objetos, permite al equipo completo contribuir en la tarea del diseño del videojuego. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. En la siguiente figura se muestra cómo se distribuye esta información.

Nombre de la clase	
Responsabilidades	Colaboradores
Atributos	
Operaciones	

Figura II.14. Card CRC XGD

Fuente: Los Autores.

Donde una clase es cualquier persona, cosa, evento, concepto o pantalla. Las responsabilidades de una clase describen a alto nivel el propósito de la existencia de la clase, normalmente una clase no debe tener más de tres o cuatro responsabilidades. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades [17].

2.1.6.11.5. UML

La realización de un proyecto mediante XGD no tiene nada que ver con la utilización o no de UML para el análisis o el diseño. En extreme Game Development se tiene los user stories para el análisis de requerimientos, pero, debido a que la mayoría de desarrolladores provienen del mundo de la programación orientada a objetos, suele ser usual utilizar UML para desarrollar los diagramas, pero esto no quiere decir que sea necesario, simplemente es una técnica de modelado de diagramas que puede ser útil o no dependiendo de las necesidades.

¹⁷ KENT BECK, *Extreme Programming Explained Change*, 2004

2.2. Videojuegos

Los videojuegos constituyen una de las modalidades que pertenecen a las TIC, sobre todo en los países del primer mundo, constituyen una actividad altamente demandada por los usuarios, comprendidos estos en las edades tempranas del desarrollo, la adolescencia y la juventud, aunque no se descartan también adultos.

Sin embargo, no puede decirse que a los videojuegos solo tengan acceso personas del mundo desarrollado, no caben dudas que estamos ante otro de los adelantos de la modernidad, que ha sido objeto de un proceso de globalización.

Los videojuegos también están presentes en los países pobres, lo que sucede que esencialmente el acceso a los mismos se encuentra más restringido. En nuestro país por ejemplo, pese a que no existe una producción de videojuegos con el fin de comercializarlos, también los niños, adolescentes, jóvenes y adultos, los emplean con una elevada frecuencia debido a que es una actividad que atrae mucho la atención de los jugadores y que genera una gran motivación hacia los mismos.

Influyen en las destrezas cognitivas de sus jugadores, en la coordinación viso-motriz, en la velocidad de reacción ante un estímulo determinado, son cuestiones ampliamente demostradas. También se afirma que constituyen una vía ideal para la educación y la instrucción, así como aquellos que tienen un contenido violento, probablemente influirá de manera negativa en la persona que lo juega.

2.2.1. Historia

Durante mucho tiempo ha sido complicado señalar cual fue el primer videojuego, principalmente debido a las múltiples definiciones que de este se han ido estableciendo, pero se puede considerar como primer videojuego el Nought and crosses, también llamado OXO, desarrollado por Alexander S. Douglas en 1952. El juego era una versión computarizada del tres en raya que se ejecutaba sobre la EDSAC (ordenador de la época) y permitía enfrentar a un jugador humano contra la máquina.

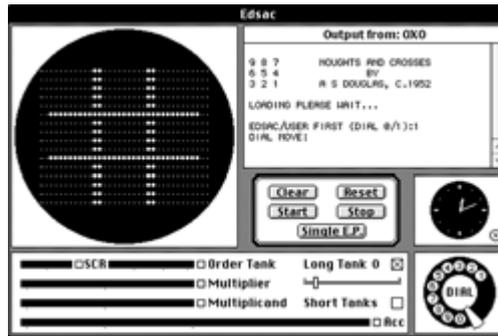


Figura II.15. El primer videojuego "Nought and crosses".

Fuente: https://encrypted-tbn2.gstatic.com/images?q=tbn:AND9GcQUsb1_cQgj0FzruftOK-cTRGraxnE028s8erWfXWBP5nhO5OSm9w

En 1957 William Higginbotham creó el juego Tennis for Two mediante un osciloscopio y una computadora analógica. El sistema era sencillo: dos barras horizontales a cada lado de una línea horizontal con una vertical en su centro dando la impresión de un campo de tenis a vista de helicóptero. Mediante el osciloscopio los jugadores hacían cálculos para golpear la supuesta bola. La empresa Atari lo desarrolló y lo llevó a la comercialización. Este fue el primer juego que permitía jugar a dos personas.

En 1966 Ralph Baer junto con Bob Tremblay crearon el primer videojuego doméstico "Fox and Hounds", que dio lugar en 1972 a la Magnavox Odyssey, la primera consola doméstica.

Durante los años '70 hubo algunos avances, pero no fue hasta 1975 cuando se consolidó el videojuego como industria, cuando Atari comercializó el "Pong" para lugares públicos. Durante los años siguientes llegaron muchos avances técnicos en los videojuegos como los microprocesadores o los chips de memoria y aparecieron más juegos en los salones recreativos y sistemas domésticos. En 1976 apareció un tipo nuevo de juego, las aventuras de texto, donde el jugador tomaba decisiones y solucionaba acertijos. Son los precursores de las aventuras gráficas. Todos estos juegos estaban desarrollados en 8 bits.

Los años '80 comenzaron con un fuerte crecimiento en el sector. Llegaron muchos sistemas domésticos (Intellivision, Colecovision, etc) y más juegos en máquinas recreativas (Pacman, Pole Position, etc). Japón apostó por las consolas como la Nintendo Entertainment System. A lo largo de la década siguieron apareciendo sistemas

domésticos y juegos tan conocidos como el Tetris. A finales de los '80 comenzaron los juegos y consolas en 16 bits, como la Mega Drive.

Un hito importante de esta década fue la aparición del fontanero más famoso del mundo, Super Mario Bros, que supuso un punto de inflexión en el desarrollo de los videojuegos. Este juego no repetía las pantallas en un bucle, sino que teníamos como objetivo llegar al final del videojuego.

Otra rama eran los videojuegos portátiles. Comenzaron a principios de los '70, pero su máxima expresión de la época llegó en 1989 con la Game Boy, que ha seguido evolucionando y mejorando hasta la actualidad.

A los '90 se llegó con la generación de los 16 bits (Mega Drive, Super Famicom, PC Engine, etc). Con esta generación se usó por primera vez el CD-ROM. Comenzaron a lanzar los primeros entornos en 3D, que acapararon la mayor parte del mercado. Se llegó a la generación de 32 bits con la Sony PlayStation, la Sega Saturn, etc.

Los videojuegos portátiles llegaron a su apogeo con la llegada de la Game Gear, Lynx, etc. Ninguna de éstas pudo hacer frente a la Game Boy y sus descendientes (GB Pocket, GB color, etc).

La PlayStation fue la consola más famosa con juegos como Final Fantasy II, Tomb Raider, Gran Turismo o Metal Gear Solid.

En 2000 Sony lanzó la PlayStation 2, Sega la Dreamcast y en 2001 se les unió Microsoft con la Xbox.

Nokia lanzó la N-Gage en 2003. Era un híbrido entre teléfono/consola portátil, pero con muy poco éxito.

Las consolas portátiles más famosas de la última década han sido sin duda la Nintendo DS, muy innovadora, y la PSP, potente y con capacidad de reproducción multimedia.

En 2005 Microsoft lanzó la Xbox 360, la primera de 256 bits, es decir, la séptima generación. Sony lanzó su consola de 256 bits, la PlayStation 3, mientras que Nintendo lanzó la Wii con una nueva forma de juego ya antes vista, utilizando sensores de movimiento.

En 2007 Apple sacó el Ipod touch y el Iphone, que se consideran también como consolas portátiles, con pantalla multitáctil.

Estas últimas consolas han tenido una evolución importante con la implantación del sistema de televisores 3D.

Rápidamente los videojuegos en 3D fueron ocupando un importante lugar en el mercado, principalmente gracias a la llamada "generación de 32 bits" en las videoconsolas. En cuanto a los PC, se crearon las aceleradoras 3D que permitían un gran salto en la capacidad gráfica de los juegos.

Si hay algo que caracteriza la industria del videojuego del siglo XXI es su transformación en una industria multimillonaria de dimensiones inimaginables pocos años antes. En 2009 la industria de los videojuegos era uno de los sectores de actividad más importantes de la economía estadounidense, y en países como España generaba más dinero que la industria de la música y el cine juntos.

2.2.2. ¿Qué es un videojuego?

Es un software creado para el entretenimiento en general y basado en la interacción entre una o varias personas por medio de un controlador¹⁸ y un aparato electrónico que ejecuta dicho videojuego; este dispositivo electrónico puede ser una computadora, una máquina arcade, una videoconsola, un dispositivo portátil (un teléfono móvil, por ejemplo) los cuales son conocidos como "plataformas". Aunque, usualmente el término "video" en la palabra "videojuego" se refiere en sí a un visualizador de gráficos rasterizados, hoy en día se utiliza para hacer uso de cualquier tipo de visualizador [¹⁹].

Igualmente se lo define como un juego cuya plataforma (gaming media) es electrónica.

En un videojuego, la dimensión reglada es sostenida por el dispositivo electrónico, y esta dimensión se presenta al jugador en forma de desafíos progresivos o emergentes. Estos desafíos forman parte de un proceso de aprendizaje, que los jugadores hacen en función de las habilidades cognitivas requeridas. Asimismo, los videojuegos presentan

¹⁸ Controlador.- Periférico de entrada que permite realizar las distintas acciones dentro de un juego.

¹⁹ <http://es.scribd.com/doc/54669992/12/Definicion-de-videojuego>

una estructura semiótica de orden simulación al, que los convierte en máquinas de signos más que en signos completos per se; no obstante lo cual pueden o no presentar un guión predefinido. (Badano, 2008)



Figura II.16. Videojuegos.

Fuente: <http://discablog.files.wordpress.com/2009/08/videojuego.jpg>

2.2.3. Principales Géneros

Un género de videojuego designa un conjunto de juegos que poseen una serie de elementos comunes.

Los videojuegos se pueden clasificar como un género u otro dependiendo de su representación gráfica, el tipo de interacción entre el jugador y la máquina, la ambientación y su sistema de juego, siendo este último el criterio más habitual a tener en cuenta. A continuación se lista los principales géneros:

Aventura: Son videojuegos en los que el protagonista debe avanzar en la trama interactuando con diversos personajes y objetos.

Disparos: En estos videojuegos el protagonista debe abrirse camino a base de disparos.

Educativos: Videojuegos cuyo objetivo es dar a conocer al usuario algún tipo de conocimiento. Su mecánica puede abarcar cualquiera de los otros géneros.

Estrategia: Se caracterizan por la necesidad de manipular a un numeroso grupo de personajes, objetos o datos para lograr los objetivos. Según su temática los hay de gestión (ya sea esta económica o social) y bélicos.

Lucha: Son videojuegos basados en el combate cuerpo a cuerpo.

Survival horror: Correspondientes al género de terror.

Plataformas: Los videojuegos de plataformas se caracterizan por tener que recorrer, saltar o escalar una serie de plataformas y acantilados, con enemigos, mientras se recogen objetos para poder completar el videojuego.

Rol: Se inspiran en los juegos de rol clásicos, donde el protagonista interpreta un papel y ha de mejorar sus habilidades mientras interactúa con el entorno y otros personajes.

Musicales: Su desarrollo gira en torno a la música.

Party games: En este género los jugadores habrán de ir avanzando por turnos en un tablero virtual e ir superando diversas pruebas de tipos muy diversos en los que compiten entre sí por llegar lo antes posible a la meta, o conseguir la máxima cantidad posible de puntos.

Simulación: Involucran al jugador en una situación simulada determinada, ya sea de gestor de un zoológico, una ciudad o una vida propia virtual.

Deportivo: Son videojuegos basados en el mundo del deporte.

Carreras: Son videojuegos en los que se conducen diferentes vehículos, ya sean reales o ficticios, para ganar en diferentes carreras. Dentro de este apartado se pueden distinguir dos variantes, arcades y simuladores.

No lineal: También conocidos como "sandbox", se caracterizan por ser videojuegos en los que el jugador puede elegir el orden de las misiones o viajar libremente por el mapa del videojuego, e interactuar con casi todo lo que este a su disposición. Estos juegos suelen ser una mezcla de disparos, luchas y carreras.

2.2.4. Características

Las características fundamentales que están presentes en todo videojuego y que lo diferencian de los demás softwares del tipo interactivo son:

- **Interacción de cualquier tipo:** Se entiende por interacción a toda acción que se ejerce recíprocamente entre dos o más objetos o agentes, en el caso del videojuego, entre el o los usuarios y el sistema que está ejecutando el juego. La interacción se da cuando el jugador es capaz de introducir órdenes al juego mediante el uso de cualquier clase de mando, que produce algún tipo de acción o modificación en el mismo. Las órdenes que puede dar el jugador van desde las más simples como presionar un botón para cambiar de imagen hasta realizar acciones como dirigir a un equipo de fútbol o controlar a un automóvil de carreras.
- **Propósito concreto:** El videojuego debe presentar algún objetivo, misión o tarea que determine lo que el jugador debe hacer dentro del mismo y también lo que no puede hacer. El objetivo de cada videojuego está estrechamente ligado al género al que este pertenece, por ejemplo, en un juego de peleas el objetivo es atacar y vencer a un oponente, mientras que en un juego de plataformas el objetivo es avanzar a lo largo de un escenario hasta alcanzar la meta. Aunque normalmente se asocia a la finalidad de un videojuego con ganar o perder, esta característica no siempre se da, principalmente en géneros como los videojuegos administrativos o simuladores de vida en donde el jugador simplemente debe llevar a cabo una actividad de forma productiva pero que no necesariamente lo lleva a vencer o alcanzar una meta.
- **Servir como un medio de entretenimiento:** Los videojuegos son programas informáticos o software cuya principal función es constituir un medio de entretenimiento para el o los usuarios. Los videojuegos además pueden presentar otros objetivos como la educación, hacer ejercicio o narrar historias pero siempre teniendo como base al entretenimiento y esto deriva de la concepción de los videojuegos como una forma de juego. De este modo existen programas informáticos que se asemejan en casi todos los aspectos a un videojuego pero que no lo son realmente porque están desempeñados con una intencionalidad

distinta. Unos de los casos más ejemplares son los simuladores de realidad virtual, que aunque se controlan y tienen objetivos similares o idénticos a los videojuegos, se diseñan como programas de entrenamiento y adiestramiento de personas para su desempeño en la vida real y no como juegos o formas de entretenimiento.

2.2.5. Videojuegos 2D

Se caracterizan por representar el mundo y sus elementos a través de figuras bidimensionales planas, el jugador puede moverse únicamente por el escenario en x o en y. Es decir, hacia arriba, abajo, derecha o izquierda. La profundidad no está implementada. Pueden subdividirse según la vista que se utilice:

- ✚ **Cenital:** Los personajes se ven como si una cámara estuviese encima de ellos y el personaje se puede mover en todas direcciones en un mismo plano.
- ✚ **Lateral:** Los personajes se ven desde un lado, y se mueve de izquierda a derecha. Existe la altura y para poder moverse en y normalmente es necesario o bien el salto, o bien el uso de escaleras u otro tipo de elementos que permitan ascender.
- ✚ **Isométrico:** Es una perspectiva en dos dimensiones que pretende simular una vista en tres dimensiones. Por dentro, el algoritmo que se encarga de dibujar el juego en la pantalla, pinta los elementos más alejados antes y los elementos más cercanos después, creando una sensación de profundidad.

Sus componentes elementales son:

Píxeles

Toda imagen que se muestra en pantalla está compuesta por puntos de diferentes colores. A estos puntos, se los denomina píxeles. El pixel es la unidad mínima de color programable. Su tamaño físico en pantalla depende de la resolución a la que se trabaje. Estas resoluciones pueden variar, pero las más comunes son 640x480, 800x600 y 1024x768.

Bitmaps.- Son imágenes estáticas formadas por píxeles codificados bajo algún patrón que conforman, como base elemental, a los demás objetos antes mencionados. Este patrón o formato de píxel indica cómo debe interpretarse cada unidad de la imagen. Algunos ejemplos de bitmaps son los archivos bmp, png, jpg, gif, entre otros.

Sprites

Son los elementos gráficos (generalmente animados) formados por uno o más bitmaps que pueden ser trasladados, rotados, escalados, y que habitualmente producen eventos en la lógica del juego al colisionar unos con otros.

Mapas

Están constituidos por porciones de bitmaps llamadas mosaicos (tiles) que forman el mundo por el cual se mueven los personajes del juego. Por lo general, se crean a partir de herramientas especialmente concebidas para tal fin.

Interfaz

Es la forma que se verán los elementos GUI y HUD, mediante los cuales el usuario interactúa con el juego

2.2.6. Partes de un videojuego

El diseño de videojuegos se ha vuelto más metódico y modular, y se le aplican los mismos principios de ingeniería que a cualquier otro producto de software. Para facilitar la reutilización de código, los videojuegos actuales están compuestos de varios módulos independientes como se observa en la **Figura II.17**.

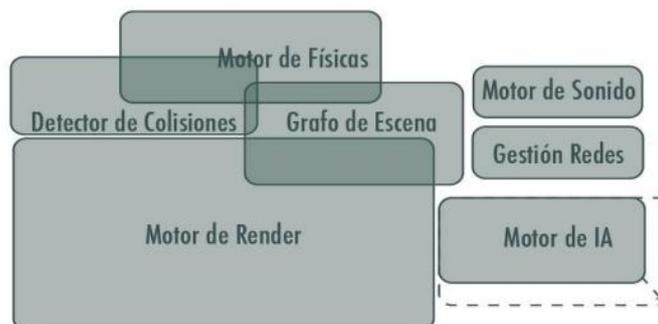


Figura II.17. Partes de un videojuego.

Fuente: <http://www.xcubo.es/2drazing/doc/Java2D.pdf>

Motor gráfico o de render.- Es el que proporciona funciones de renderizado 2D, 3D y de sprites. Se suele apoyar en librerías o API's, Define una capa de abstracción entre el hardware/software y el juego en sí. Además, se encarga de trabajar con lo que es visible y lo que no.

Grafo de escena.- Es una estructura de datos muy utilizada en editores gráficos basados en vectores y en videojuegos. Se encarga de ordenar la representación lógica y, también a veces, espacial de una escena gráfica.

Motor de físicas.- Es software que simula modelos de física utilizando variables del tipo velocidad, masa, etc. Su objetivo es simular y predecir los efectos bajo diversas situaciones de lo que ocurre en la vida real o en un mundo de fantasía. En los juegos se utilizan cada vez más, sobre todo en los de carreras donde factores como la fuerza centrífuga o derrapes necesitan de ecuaciones físicas para aportar realismo.

Detector de colisiones.- Es un algoritmo empleado para detectar cuándo dos o más objetos colisionan. Se suelen realizar mediante el cálculo de la intersección de volúmenes simples (Bounding Volumes). Se pueden utilizar distintos volúmenes para envolver el objeto y así detectar mejor las colisiones, como cubos o esferas. Se puede tomar como un componente independiente, aunque tiene mucho que ver con el grafo de escena y el motor de físicas.

Motor de inteligencia artificial.- Es el encargado de dotar a ciertos elementos del juego un comportamiento pseudo-inteligente. Por regla general, lo que se aplica en los juegos son técnicas muy simples de inteligencia artificial como máquinas de estados o algoritmos de búsquedas.

Motor de sonido.- Es el encargado de reproducir la banda sonora del juego y los efectos de sonido, como disparos o explosiones, en sincronía con la acción del juego.

Gestión de redes.- Está alcanzando una progresiva importancia. Hoy por hoy, casi todos los juegos tienen un componente de red. Existen juegos exclusivamente online, de los cuales, la mayoría ofrece partidas multijugador a través de red.

2.2.7. Estructura interna de un videojuego

Durante la ejecución de un juego, siempre hay un bucle ejecutando lo que es necesario durante todo el juego, en la **Figura II.18.**, se observa cuál es la estructura básica de un videojuego:

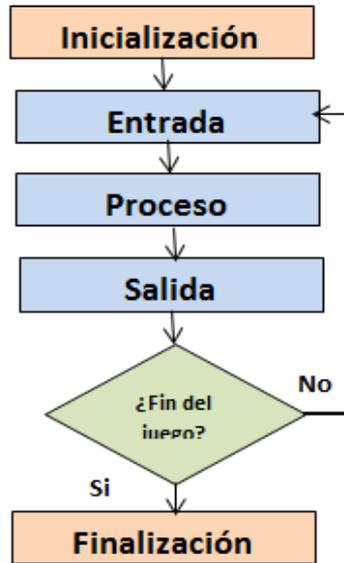


Figura II.18. Estructura de un videojuego.

Fuente: Los Autores.

La estructura de un videojuego se puede dividir en tres grandes partes. La primera es la inicialización, la segunda el bucle del juego (separado en tres partes más) y la tercera la finalización. Cada una de ellas tiene sus propias particularidades:

- **Iniciación:** Es la parte del juego donde se ponen los valores iniciales de las variables y se cargan todos los recursos externos necesarios.

- **Bucle del juego:** Se divide a su vez en tres partes, como son:
 - **Entrada:** Donde se capturan las interacciones del usuario, ya sea por ratón, joystick, teclado.
 - **Proceso:** Donde se procesa la información que le llega en el jugador, se pone en marcha la lógica interna de cada una de las partes del juego, se calcula la IA y las físicas.

- **Salida:** Muestra la respuesta, si hay sonidos también se emiten en esta parte.

➤ **Finalización:** Se liberan los recursos, se guardan los récords en caso de haber.

Todo videojuego, por más sencilla o complejo, sigue esta estructura básica para su creación.

2.3. ANDROID



Figura II.19. Logo Android.

Fuente: <http://dontfearthetech.blogspot.com/2012/05/what-is-android.html>

En los últimos años los teléfonos móviles han experimentado una gran evolución, desde los primeros terminales, grandes y pesados, pensados sólo para hablar por teléfono en cualquier parte, a los últimos modelos, con los que el término “medio de comunicación” queda pequeño.

Es así como nace Android. Android es una plataforma software y un sistema operativo basado en Linux, diseñado originalmente para dispositivos móviles, tales como teléfonos inteligentes, pero que posteriormente se expandió su desarrollo para soportar otros dispositivos tales como tablets, reproductores MP3, netbooks, PCs e incluso televisores. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas

aplicaciones, widgets²⁰, o incluso, modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma.

2.3.1. Arquitectura

Android es un sistema operativo libre basado en el kernel de Linux que se encarga de los servicios bases del sistema como seguridad, gestión de memoria, gestión de procesos etc. El núcleo también actúa como capa de abstracción entre el hardware y el resto del software [²¹]. En la **Figura II.20.**, se observa el núcleo donde se superponen las siguientes capas:

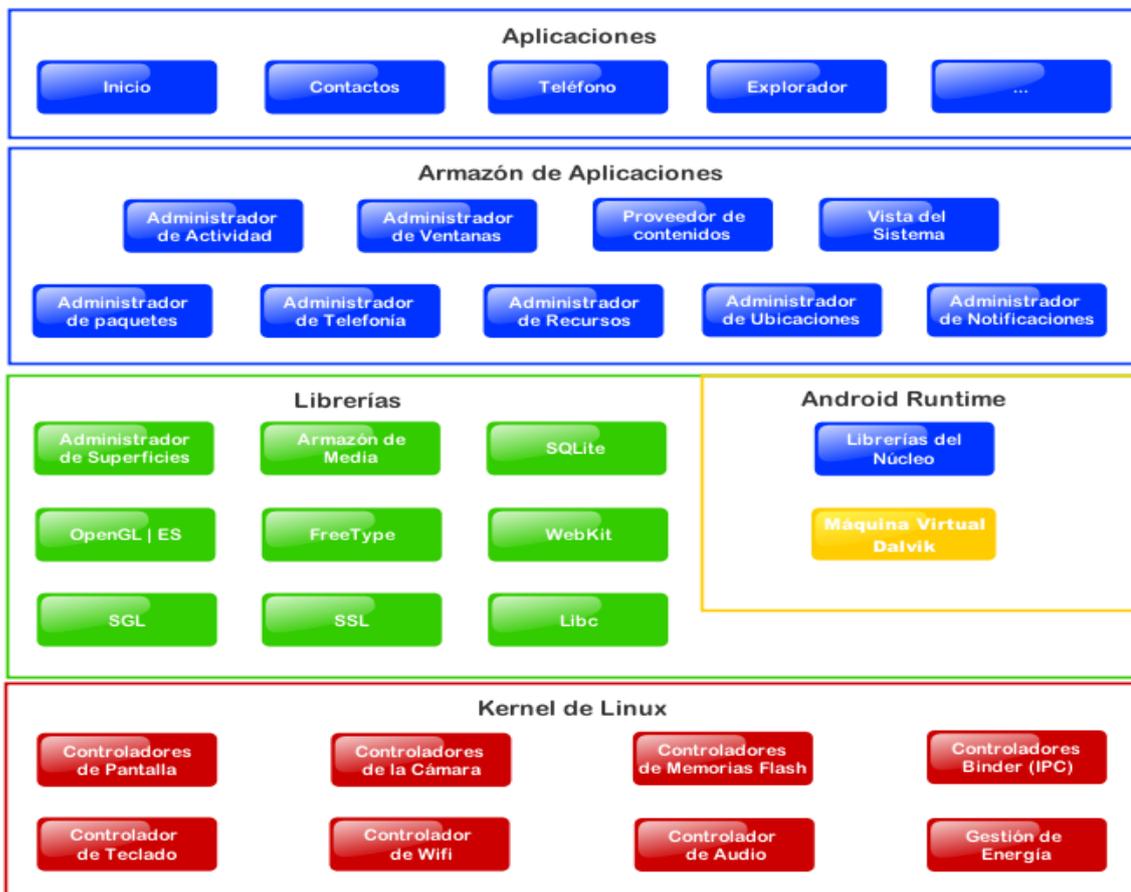


Figura II.20. Arquitectura Interna del conjunto de software Android.

Fuente: <http://2.bp.blogspot.com/-LpURpEBykA8/Tm-vMLZSR9I/AAAAAAAAABw/ySehVzq8W6w/s400/android-aplicaciones.jpg>

²⁰ Un widget es una pequeña aplicación que facilita el acceso a funciones frecuentes.

²¹ <http://www.xatakandroid.com/sistema-operativo/que-es-android>
<http://www.configurarequipo.com/doc1107.html>

- **Kernel de Linux:** Android se basa en la versión 2.6 del kernel (núcleo) de Linux que actúa como una capa de abstracción entre el hardware y el resto del conjunto de software, además de prestar los servicios de seguridad, gestión de memoria, gestión de procesos, network stack, driver model.
- **Librerías:** Incluye un conjunto de librerías de C / C + + utilizado diversos componentes del sistema. Estas capacidades están expuestas a los desarrolladores a través de la aplicación marco. Algunas de las principales librerías se enumeran a continuación:
 1. Sistema de librería de C un derivado de BSD aplicación del sistema de biblioteca estándar para Linux sintonizado los dispositivos embebido basados en C (libc)
 2. Medios de comunicación de librerías IDE basadas en OpenCORE
 3. PacketVideo, para la reproducción y la grabación de formatos de audio y vídeo, así como archivos de imagen, MPEG4, H.264, MP3, AAC, AMR, JPG y PNG, ogg vorbis y ogg theora.
 4. Superficie Manager – gestiona el acceso a la pantalla y a los subsistemas compuestos 2D y gráficos 3D a partir de capas múltiples aplicaciones
 5. LibWebCore - un moderno navegador web
 6. SGL - el motor de gráficos 2D subyacente
 7. Librerías 3D basada en OpenGL ES 1.0 API
 8. FreeType - mapa de bits y vectores de la renderización de fuentes
 9. SQLite – un ligero y potente motor de bases de datos relacionales disponibles para todas las aplicaciones.
- **Android Runtime:** Android incluye un conjunto de librerías que proporciona la mayor parte de la funcionalidad disponible en el núcleo de las bibliotecas el lenguaje de programación Java. Cada aplicación se ejecuta en su propio proceso,

con su propia instancia de la máquina virtual Dalvík que se ha escrito de modo que un dispositivo puede ejecutar múltiples máquinas virtuales de manera eficiente. La máquina virtual de Dalvík ejecuta en el Dalvík archivos ejecutables (.Dex) que el formato está optimizado para memoria mínima. La máquina virtual se basa en registros, y corre clases compiladas por un compilador de lenguaje Java que se han transformado en el. Dex incluido el formato de la “dx” herramienta. La máquina virtual de Dalvík se basa en el kernel de Linux para la funcionalidad subyacente como subprocessos y de bajo nivel de gestión de memoria.

➤ **Armazón de Aplicaciones:** Esta capa está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik, las mismas son:

- *Activity Manager:* Se encarga de administrar la pila de actividades de nuestra aplicación así como su ciclo de vida.
- *Windows Manager:* Se encarga de organizar lo que se mostrará en pantalla. Básicamente crea las superficies en la pantalla que posteriormente pasarán a ser ocupadas por las actividades.
- *Content Provider:* Esta librería es muy interesante porque crea una capa que encapsula los datos que se compartirán entre aplicaciones para tener control sobre cómo se accede a la información.
- *Views:* En Android, las vistas son los elementos que nos ayudarán a construir las interfaces de usuario: botones, cuadros de texto, listas y hasta elementos más avanzados como un navegador web o un visor de Google Maps.
- *Notification Manager:* Engloba los servicios para notificar al usuario cuando algo requiera su atención mostrando alertas en la barra de estado. Un dato importante es que esta biblioteca también permite jugar con sonidos, activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos.

- *Package Manager*: Esta biblioteca permite obtener información sobre los paquetes instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes. Con paquete nos referimos a la forma en que se distribuyen las aplicaciones Android, estos contienen el archivo .apk, que a su vez incluyen los archivos .dex con todos los recursos y archivos adicionales que necesite la aplicación, para facilitar su descarga e instalación.
 - *Telephony Manager*: Con esta librería podremos realizar llamadas o enviar y recibir SMS/MMS, aunque no permite reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso.
 - *Resource Manager*: Con esta librería podremos gestionar todos los elementos que forman parte de la aplicación y que están fuera del código, es decir, cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos o layouts. En un post relacionado a la estructura de un proyecto Android veremos esto más a fondo.
 - *Location Manager*: Permite determinar la posición geográfica del dispositivo Android mediante GPS o redes disponibles y trabajar con mapas.
 - *Sensor Manager*: Nos permite manipular los elementos de hardware del teléfono como el acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.
 - *Cámara*: Con esta librería podemos hacer uso de la(s) cámara(s) del dispositivo para tomar fotografías o para grabar vídeo.
 - *Multimedia*: Permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.
- **Aplicaciones**: Se suministran con un conjunto de aplicaciones que incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos, y muchas más que puedes obtener desde el sitio Market Android.

2.3.2. Características

❖ **Conectividad:**

- ✓ Navegador integrado: basado en el motor open source WebKit.
- ✓ Soporte telefonía GSM, depende del hardware del teléfono.
- ✓ Bluetooth, EDGE, 3G y WIFI, depende del hardware del teléfono.

❖ **Software:**

- ✓ Gráficos optimizados, con una librería de gráficos 2D; gráficos 3D basado en la especificación OpenGL ES 1.0 (aceleración por hardware opcional).
- ✓ SQLite para el almacenamiento de datos estructurados (Bases de datos)
- ✓ Ambiente rico de desarrollo incluyendo un emulador de dispositivo, herramientas para depurar, perfiles de memoria y rendimiento, y un plugin para el IDE Eclipse.
- ✓ Framework de aplicaciones: permite reuso y reemplazo de componentes.
- ✓ Máquina virtual Dalvik: optimizada para dispositivos móviles.
- ✓ Android Market permite que los desarrolladores pongan sus aplicaciones, gratuitas o de pago, en el mercado a través de esta aplicación accesible desde todos los teléfonos con Android.
- ✓ Entorno de desarrollo SDK que proporciona un plugin para el IDE de Eclipse y APIs necesarias para empezar a desarrollar aplicaciones en la plataforma Android usando un lenguaje de programación java el cual incluye un emulador de dispositivo, herramientas para la depuración, memoria y rendimiento de perfil.

❖ **Hardware y multimedia**

- ✓ Soporte para medios con formatos comunes de audio, vídeo e imágenes planas (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- ✓ Cámara, GPS, Brújula, acelerómetro, depende del hardware del Teléfono.
- ✓ Pantalla táctil.

2.3.3. Ventajas para el Desarrollo de Aplicaciones

Existen muchas ventajas para el desarrollo de aplicaciones para esta plataforma como son:

- ✓ **Cero costos de inicio para comenzar el desarrollo:** Las herramientas de desarrollo para la plataforma son de descarga gratuita, y Google sólo cobra una pequeña cuota para distribuir las aplicaciones en el Android Market.
- ✓ **Libertad para innovar:** Además de la creación de aplicaciones para funcionar en dispositivos Android, los desarrolladores también son libres de contribuir o ampliar la plataforma.
- ✓ **Libertad para colaborar:** Los desarrolladores de Android no están obligados a firmar un acuerdo de confidencialidad y se les anima a colaborar y compartir el código fuente de uno con el otro. Según una encuesta realizada por el software de Pato Negro, el número de aplicaciones de código abierto y bibliotecas móviles creció a una tasa de 168% entre 2008 y 2009, más rápido en Android que cualquier otra plataforma. Esto significa más código que se puede reutilizar en sus propios proyectos para llevar al mercado mucho más rápido.
- ✓ **Modelo de distribución abierta:** Muy pocas restricciones se colocan en el contenido o funcionalidad permitido en Android Market de Google, y los desarrolladores tienen la libertad de distribuir sus aplicaciones a través de otros canales de distribución también.
- ✓ **Multiplataforma de apoyo:** Hay una amplia variedad de dispositivos de hardware que funcionan con el sistema operativo Android, entre ellos muchos de los teléfonos diferentes, PCs y Tablet. El desarrollo de la plataforma se puede producir en el sistema operativo Windows, Mac o Linux.
- ✓ **Multi-portadora de apoyo:** Un gran número de operadores de telecomunicaciones que actualmente ofrecen los teléfonos Android potencia

2.3.4. Requisitos que debe cumplir una terminal Android

Un terminal Android a nivel de hardware debe cumplir con los siguientes requisitos:

- ✓ Debe tener una resolución de las siguientes:
 - 240x320
 - 240x400
 - 240x432
 - 320x480
 - 480x800
 - 480x854
- ✓ Acelerómetro para cambiar la orientación de la pantalla entre landscape y portrait
- ✓ Pantalla táctil con tecnología resistiva o capacitiva, prefiriendo las segundas que soportan tecnología multitáctil.
- ✓ Puerto micro USB que implemente un cliente para almacenamiento masivo y un ADB (Android Debug Bridge).
- ✓ Teclas de navegación (atrás, menú, home y buscar).
- ✓ Soporte para tecnologías Wifi y 3G o HSPA.
- ✓ Cámara de al menos 2 MegaPixels preferiblemente con autofocus.
- ✓ Un acelerómetro de 3 ejes capaz de actualizarse a 50Hz o más.
- ✓ Una brújula digital de 3 ejes capaz de actualizarse a 10Hz o más.
- ✓ Antena GPS o aGPS (GPS asistido, para mejorar la precisión en lugares con señal GPS débil).
- ✓ Bluetooth

2.3.5. La máquina virtual Dalvik

Esta máquina virtual ha sido específicamente diseñada para Android, optimizada y adaptada a las peculiaridades propias de los dispositivos móviles (menor capacidad de proceso, baja memoria, alimentación por batería, etc.) y trabaja con ficheros de extensión .dex (Dalvik Executables). Dalvik no trabaja directamente con el bytecode de Java, sino que lo transforma en un código más eficiente que el original, pensado para procesadores pequeños.

Gracias a la herramienta “dx”, esta transformación es posible: los ficheros .class de Java se compilan en ficheros .dex, de forma que cada fichero .dex puede contener varias clases. Después, este resultado se comprime en un único archivo de extensión .apk (Android Package), que es el que se distribuirá en el dispositivo móvil. Dalvik permite varias instancias simultáneas de la máquina virtual, y a diferencia de otras máquinas virtuales, está basada en registros y no en pila, lo que implica que las instrucciones son más reducidas y el número de accesos a memoria es menor. Así mismo, Dalvik no permite la compilación Just-in-Time.

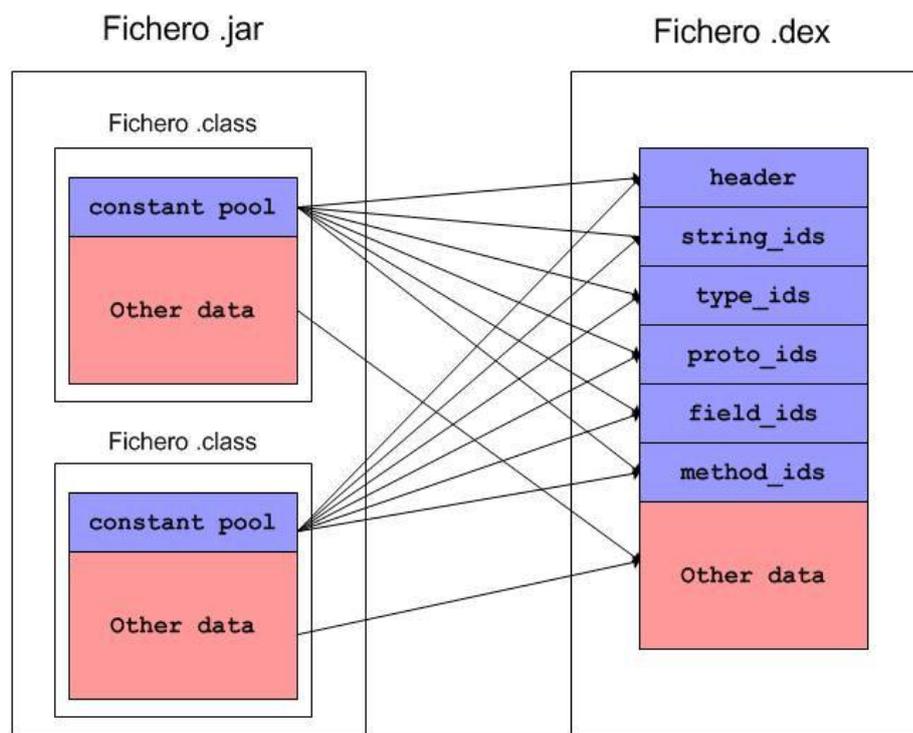


Figura II.21. Formato de un fichero .dex

Fuente: <https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcSeRYpm7N8nZWtlqmJNYFTvtgtj5A2s7ykyA5xZbd15K-1PcjEbvww>

Según los responsables del proyecto, la utilización de esta máquina virtual responde a un deseo de mejorar y optimizar la ejecución de aplicaciones en dispositivos móviles, así como evitar la fragmentación de otras plataformas como Java ME. A pesar de esta aclaración oficial, no ha pasado inadvertido que con esta maniobra Android ha esquivado tener que utilizar directamente Java ME, evitando así que los futuros desarrolladores de aplicaciones tengan que comprar ninguna licencia, ni tampoco que publicar el código fuente de sus productos. El lenguaje Java utilizado en Android no sigue ningún JSR determinado, y Google se afana en recordar que Android no es tecnología Java, sino que simplemente utiliza este lenguaje en sus aplicaciones [22].

2.3.6. Entorno de desarrollo de Android

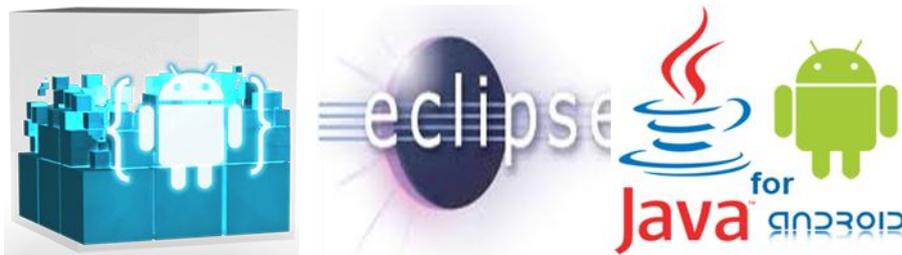


Figura II.22. Entorno de Desarrollo de Android

Fuente: http://www.smartblog.es/wp-content/uploads/2010/11/eclipse_android_sdk.png

Una de las ventajas de este entorno es que se puede instalar en Windows, en Linux y en Mac O.S. X, lo único que se necesita para desarrollar una aplicación es el SDK de Android. Un kit que incluye una serie de herramientas como son: el Traceview, herramienta que nos permite extraer estadísticas de rendimiento de la aplicación; el LogCat, que nos presenta los mensajes que se imprimen desde el código durante la ejecución de una aplicación; y herramientas para generar los instaladores de las aplicaciones que desarrollemos; entre otras herramientas de utilidad.

Además, el SDK incluye la API de Android, unas librerías que contienen todas las clases y operaciones que debemos utilizar para poder comunicarnos con el S.O. y de esta forma poder, por ejemplo, definir una actividad.

²² <http://gdroid.com.mx/blog/2011/06/12/la-maquina-virtual-dalvik/>

En cuanto al IDE o entorno que integra las herramientas necesarias para desarrollar aplicaciones, se puede utilizar cualquiera. Incluso se podría utilizar un editor de textos normal para programar una aplicación y no utilizar un IDE. Aun así, es recomendable el uso de Eclipse, debido a que existe un *plugin* oficial llamado ADT que una vez instalado en Eclipse nos permite acceder a la API y a las diferentes herramientas que el SDK nos brinda directamente desde la interface de Eclipse, facilitando enormemente el desarrollo de las aplicaciones.

En cuanto al lenguaje de programación, la API se encuentra en Java y el lenguaje recomendado para programar es este. Sin embargo, el uso de C sigue siendo imprescindible para desarrollar con éxito algunas partes de un videojuego complejo, como la gestión de físicas. Igualmente a través del paquete NDK, podemos utilizar funciones programadas en C, que se ejecutarán de forma nativa durante la ejecución de la aplicación. Claro está que, al ser C un lenguaje no interpretado, se debe compilar este código para cada una de las arquitecturas donde la aplicación de Android vaya a ejecutarse. Para facilitar las cosas, Android permite introducir en el mismo instalador de la aplicación varias compilaciones diferentes de las operaciones en C, para que el mismo instalador funcione en varias arquitecturas al mismo tiempo.

2.3.7. Aplicaciones

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit, pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varios cruz aplicaciones de la plataforma web móvil marcos, y también es posible usar las bibliotecas Qt gracias al proyecto Necesitas SDK.

Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos

Google Play

Google Play es la tienda en línea de software desarrollado por Google para dispositivos Android. Una aplicación llamada "play store" que se encuentra instalada en la mayoría

de los dispositivos Android y permite a los usuarios navegar y descargar aplicaciones publicadas por los desarrolladores. Google retribuye a los desarrolladores el 70% del precio de las aplicaciones.

Por otra parte, los usuarios pueden instalar aplicaciones desde otras tiendas virtuales (tales como Amazon Appstore o SlideME) o directamente en el dispositivo si se dispone del archivo APK de la aplicación.

Privacidad

Se han descubierto ciertos comportamientos en algunos dispositivos que limitan la privacidad de los usuarios, de modo similar a iPhone, pero ocurre al activar la opción “Usar redes inalámbricas” en el menú “Ubicación y seguridad”, avisando que se guardarán estos datos, y borrándose al desactivar esta opción, pues se usan como caché y no como log tal como hace iPhone.

Seguridad

Según un estudio de Symantec de 2011, comparado con el iOS, Android es un sistema menos vulnerable, debido principalmente a que el proceso de certificación de aplicaciones es más riguroso que el de Apple sin dejar al lado que es un sistema libre y el esquema de gestión de permisos para estas, que pone muchas decisiones de seguridad importantes en manos del usuario, lo expone a mayores riesgos y le hace más atractivo a los ataques de ingeniería social. Sin embargo, basándose en números absolutos contabilizados de los boletines de seguridad de Microsoft, Adobe, Oracle y Apple, el 85% de los agujeros de seguridad en 2011 y 2012 los mayores ataques fueron para el sistema iOS, pese a que este posee una cuota de mercado inferior a la de Android.

2.3.8. Estructura de una Aplicación

Las aplicaciones de Android se estructuran en componentes, cada componente de una aplicación juega un papel específico dentro de esta y existe por sí mismo. Eso sí, puede haber componentes que dependan unos de otros.

Es decir, a la hora de desarrollar una aplicación para Android no tendremos una sola función principal (o función *main*) la cual lanzará las diferentes pantallas de la

aplicación. Sino que tendremos varios componentes independientes cuya comunicación será llevada a través del S.O.

Además, estos componentes, lejos de tener cada uno una sola función *main* la cual nos permitiría llevar a cabo una ejecución secuencial de la aplicación, implementan una serie de funciones que serán llamadas por el S.O. cuando se cumpla la condición necesaria para llamar a cada función. Por tanto, las aplicaciones en Android funcionan de forma asíncrona y es el S.O. el que ante las peticiones del usuario va llamando a una y otra función de los diferentes componentes de la aplicación, según convenga.

A través de la API de Android podemos declarar cuatro tipos de componentes:

- Actividades (Activities).
- Servicios (Services).
- Proveedores de Contenido (Content Providers).
- Receptores de Transmisión (Broadcast Receivers).

Las **Actividades** son los componentes más importantes, de hecho los videojuegos desarrollados se componen exclusivamente de este tipo de componentes. Una actividad es una pantalla de la aplicación, con su interface de usuario, es decir, lo que en el desarrollo de aplicaciones se conoce habitualmente como *vista*.

Un **Servicio**, en cambio, es un componente que no tiene interface propia, sino que se ejecuta en segundo plano y lleva a cabo tareas largas que consuman muchos recursos. Por ejemplo, en caso de querer implementar un reproductor de música, este tendrá un servicio que reproduce esta música. Al mismo tiempo, la aplicación tendrá una actividad que permite que el usuario escoja la canción deseada. La actividad se comunicará con este servicio para pausar, reanudar o cambiar de canción.

En cuanto al **Proveedor de Contenido**, este es un tipo de componente que permite almacenar una información para que después pueda ser accedida desde varias aplicaciones. Según cómo configuramos al Proveedor limitaremos dicho contenido a unas aplicaciones muy concretas o no. Se trata del único mecanismo para compartir información entre diferentes aplicaciones que existe en Android.

Por último, se encuentra el **Receptor de Transmisión**, este crea una notificación que se presentará en la Barra de Notificaciones del sistema. Normalmente, una notificación contiene una información acerca de lo que se pretende informar y un apuntador a una actividad o servicio que será lanzado cuando se presione sobre la notificación.

Hay que tener en cuenta que estos componentes son representados por clases Java, lo que significa que, cuando la aplicación se ejecuta, se crean instancias de estos, logrando crear más de una instancia del mismo componente. Dichas instancias además son independientes.

2.3.9. Ciclo de vida de las Aplicaciones Android

En Android, cada aplicación se ejecuta en su propio proceso. Esto aporta beneficios en cuestiones básicas como seguridad, gestión de memoria, o la ocupación de la CPU del dispositivo móvil. Android se ocupa de lanzar y parar todos estos procesos, gestionar su ejecución y decidir qué hacer en función de los recursos disponibles y de las órdenes dadas por el usuario.

Android lanza tantos procesos como permitan los recursos del dispositivo. Cada proceso, correspondiente a una aplicación, está formado por una o varias actividades independientes (componentes Activity) de esa aplicación. Cuando el usuario navega de una actividad a otra, o abre una nueva aplicación, el sistema duerme dicho proceso y realiza una copia de su estado para poder recuperarlo más tarde. El proceso y la actividad siguen existiendo en el sistema, pero están dormidos y su estado ha sido guardado. Es entonces cuando crea, o despierta si ya existe, el proceso para la aplicación que debe ser lanzada, asumiendo que existan recursos para ello.

Cada uno de los componentes básicos de Android tiene un ciclo de vida bien definido; esto implica que el desarrollador puede controlar en cada momento en qué estado se encuentra dicho componente, pudiendo así programar las acciones que mejor convengan. El componente Activity, probablemente el más importante, tiene un ciclo de vida como se observa en la **Figura II.23**.

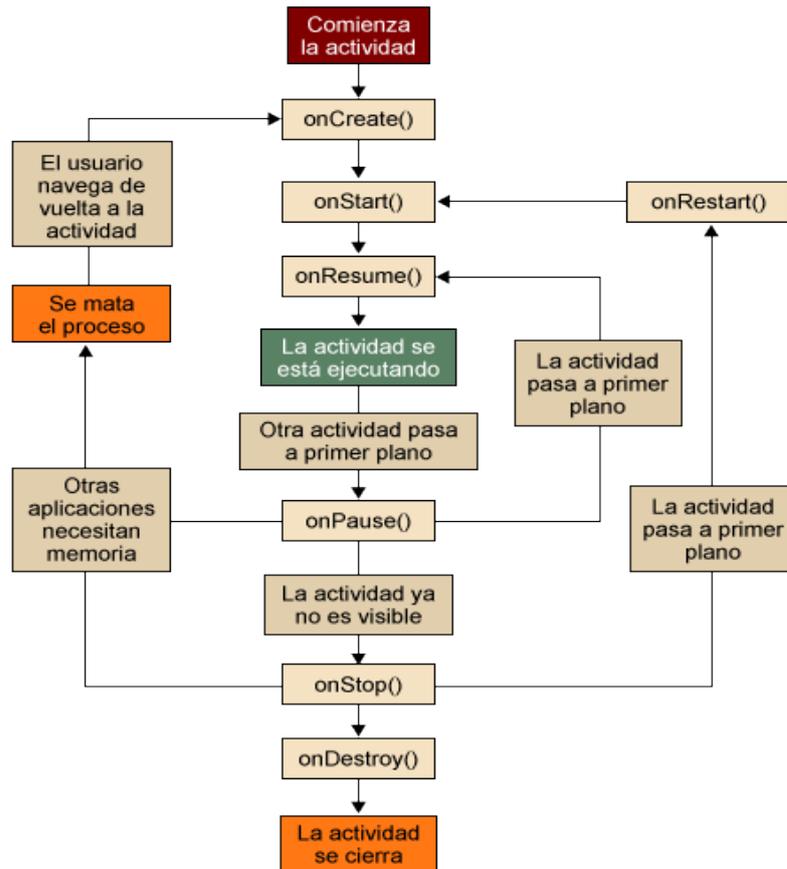


Figura II.23. Ciclo de vida de un objeto Activity.

Fuente: http://cv.uoc.edu/~mgalicia/practica_final/img/modulo5_2_img3.gif

A continuación se describen las operaciones a través de las cuales se gestiona el ciclo de vida de una actividad:

- **onCreate()**, **onDestroy()**: abarcan todo el ciclo de vida. Cada uno de estos métodos representan el principio y el fin de la actividad.
- **onStart()**, **onStop()**: representan la parte visible del ciclo de vida. Desde onStart() hasta onStop(), la actividad será visible para el usuario, aunque es posible que no tenga el foco de acción por existir otras actividades superpuestas con las que el usuario está interactuando. Pueden ser llamados múltiples veces.
- **onResume()**, **onPause()**: delimitan la parte útil del ciclo de vida. Desde onResume() hasta onPause(), la actividad no sólo es visible, sino que además tiene el foco de la acción y el usuario puede interactuar con ella.

CAPÍTULO III

3. ANÁLISIS Y ADAPTACIÓN DE LAS METODOLOGÍAS ÁGILES SCRUM Y EXTREME GAME DEVELOPMENT EN UNA METODOLOGÍA PARA EL DESARROLLO DE VIDEOJUEGOS

Este capítulo se centra en la determinación de parámetros a través de los criterios “Planificación”, “Ciclo de Vida del Proyecto”, “Calidad” y “Herramientas Utilizadas en el Desarrollo”; para analizar las metodologías ágiles Scrum y Extreme Game Development, mediante el desarrollo de un prototipo de videojuego con cada una de las mismas.

Luego de obtenerse los resultados se lleva a cabo la adaptación de las metodologías en una nueva metodología para desarrollo de videojuegos en la plataforma Android.

3.1. Determinación de Criterios para el Análisis

Los criterios que se van a considerar en general para analizar las metodologías ágiles Scrum y Extreme Game Development mediante el desarrollo de un prototipo de videojuego, se muestra en la **Tabla N° III.I.**, los mismos, han sido tomados como base de la referencia [²³] y por los autores de esta tesis.

Tabla III.I. Determinación de los Criterios para el Análisis.

CRITERIOS	DESCRIPCIÓN
Planificación	Se considera como organiza cada metodología las reuniones y como son distribuidos los roles en el proyecto, además que valores promueven.
Ciclo de vida del proyecto	Se consideran las fases, actividades, prácticas y artefactos de cada metodología como parte de su ciclo de vida en un proyecto
Calidad	Se busca analizar si las metodologías contemplan ciertos parámetros de calidad en su enunciado metodológico.
Herramientas utilizadas en el desarrollo	Se da a conocer las distintas herramientas que las metodologías poseen para cumplir las tareas que especifican sus enunciados.

Elaborado por: Los Autores.

Los cuatro criterios generales que se ha tomado en cuenta para el análisis están divididos en varios parámetros que detallamos a continuación:

²³ JOSÉ CARLOS CARVAJAL RIOLA, *Metodologías ágiles*, 2008

Tabla III.II. Parámetros del Criterio “Planificación” para el Análisis.

PLANIFICACIÓN	
PARÁMETRO	DESCRIPCIÓN
Roles	Distribución de funciones entre los miembros del proyecto, donde se definen las tareas de cada uno.
Reuniones	Junta de los miembros para planificar, discutir y analizar problemas en el desarrollo del proyecto.
Valores	Promover a cada miembro del equipo ciertos valores en su comportamiento con el propósito de llevar con éxito el proyecto.

Elaborado por: Los Autores.

Tabla III.III. Parámetros del Criterio “Ciclo de vida del proyecto” para el Análisis.

CICLO DE VIDA DEL PROYECTO	
PARÁMETRO	DESCRIPCIÓN
Fases	El progreso del desarrollo del software, que contienen los procesos, las actividades y las tareas involucradas en la realización del videojuego, desde la definición de los requisitos hasta la finalización.
Actividades	Conjunto de acciones planificadas que consisten en la ejecución de ciertos procesos o tareas que se llevan a cabo para crear un videojuego de calidad.
Prácticas	Acciones con repercusiones tangibles para mejorar el desarrollo del proyecto.
Artefactos	Instrumentos empleados para el desarrollo del proyecto que sirven para comprender mejor tanto el análisis como el diseño del videojuego (entre otros),

Elaborado por: Los Autores.

Tabla III.IV. Parámetros del Criterio “Calidad” para el Análisis.

CALIDAD	
PARÁMETRO	DESCRIPCIÓN
Fiabilidad	Trazabilidad entre los artefactos producidos en las distintas etapas del ciclo de vida del proyecto y simplicidad de los diseños, en la implementación y desarrollo del videojuego en general.
Usabilidad	Documentación clara y precisa
Mantenibilidad	Si la metodología promueve que los videojuegos desarrollados bajo su enfoque sean simples al momento de mantenerse.

Elaborado por: Los Autores.

Tabla III.V. Parámetros del Criterio “Herramientas utilizadas en el desarrollo” para el Análisis.

HERRAMIENTAS UTILIZADAS EN EL DESARROLLO	
PARÁMETRO	DESCRIPCIÓN
Gestión, medición y seguimiento	La forma de llevar un control de lo que se está realizando en el proyecto.
Refactorización	Modificación del código del videojuego sin cambiar su comportamiento para mejorar la facilidad de comprensión o eliminar código muerto.
Integración	Compilación y ejecución de tests de todo el proyecto lo más a menudo posible para poder detectar fallos cuanto antes

Elaborado por: Los Autores.

3.2. Escala de Evaluación

La forma para evaluar las dos metodologías en base a los cuatro criterios antes mencionados y así obtener resultados cuantitativos y cualitativos que nos permitan escoger las partes más importantes de cada metodología analizada, para adaptarlas en la nueva metodología para desarrollo de videojuegos, se indican en la siguiente tabla:

Tabla III.VI. Escala de Evaluación para las Metodologías.

VARIABLE	EQUIVALENCIA	PUNTAJE
EA	Evaluación Mala	0
EB	Evaluación Regular	1
EC	Evaluación Bueno	2
ED	Evaluación Muy Bueno	3
EE	Evaluación Excelente	4

Elaborado por: Los Autores.

Tabla III.VII. Pesos en el criterio para los parámetros.

DESCRIPCIÓN	PESO
PP1	= [EE]
PP2	= [EE]
...
PP4	= [EE]

Elaborado por: Los Autores.

Dónde:

PP1, PP2,..., PPi = Peso más alto para los parámetros en las dos metodologías en el criterio.

El valor de cada parámetro en su respectivo criterio de interpretación para la metodología incluye la siguiente fórmula:

$$P1 \geq 0 \text{ y } P1 \leq PP1$$

$$P2 \geq 0 \text{ y } P2 \leq PP2$$

$$Pi \geq 0 \text{ y } Pi \leq PPi$$

Dónde:

Pi = Valor de cada parámetro en el criterio.

La calificación definitiva de la metodología en base a cada criterio del análisis se obtiene sumando los puntajes obtenidos, utilizando las siguientes fórmulas:

$$Pascrum = \Sigma (Pi)$$

$$Paxgd = \Sigma (Pi)$$

$$Pc = \Sigma (PPi)$$

Dónde:

Pascrum = Puntaje acumulado por Scrum en el criterio.

Paxgd = Puntaje acumulado por XGD en el criterio.

Pc = Puntaje sobre el que se califica el criterio.

Tabla III.VIII. Interpretación de Resultados en cada Metodología.

DESCRIPCIÓN	RESULTADO
Cscrum	= (Pascrum / Pc) * 100%
Cxgd	= (Paxgd / Pc) * 100%

Elaborado por: Los Autores.

Dónde:

Cscrum = Calificación que obtuvo la metodología Scrum en el criterio.

Cxgd = Calificación que obtuvo la metodología XGD en el criterio

Tabla III.IX. Calificación para las Metodologías.

DESCRIPCIÓN	CALIFICACIÓN
Malo	< 60%
Regular	>= 60% y <=70 %
Bueno	> 70% y <=80%
Muy bueno	> 80% y <= 90 %
Excelente	> 90 %

Elaborado por: Los Autores.

3.3. Prototipo de videojuego para el Análisis

La realización del análisis para las dos metodologías se llevará a cabo en el siguiente escenario que se basa en el juego “ARKANOID”, en donde el jugador controla una pequeña plataforma o pala, conocida como "Nave Espacial Vaus", que impide que una bola salga de la zona de juego, haciéndola rebotar. En la parte superior hay bloques, que desaparecen al ser tocados por la bola. Cuando no queda ningún bloque, el jugador gana y termina el juego, si no logra romper todos los bloques y ha agotado sus 3 vidas, el juego termina y el jugador ha perdido.

3.3.1. Herramientas utilizadas para el desarrollo del prototipo

Para la realización del prototipo se utilizarán las mismas herramientas en las dos metodologías, las cuales son:

- Kit de Desarrollo de Software de Android.
- ADT (Android Developer Tools).
- Entorno de desarrollo Integrado Eclipse.

3.4. Prototipo “Metodología Ágil Scrum”

3.4.1. Fase Pre-game

Roles

Para el desarrollo del prototipo, los roles fueron distribuidos entre los dos autores de esta tesis, como se muestra en la siguiente tabla:

Tabla III.X. Roles asignados para el desarrollo del prototipo con Scrum.

Persona	Contacto	Rol
Gabriel Armendáriz	garmy121t@yahoo.es / 0987382825	Scrum Master
Milton Saltos	milton3552@hotmail.com / 0984752026	Product Owner
Gabriel Armendáriz	garmy121t@yahoo.es / 0987382825	Scrum Team (Artista)
Milton Saltos	milton3552@hotmail.com / 0984752026	Scrum Team (Diseñador)
Gabriel Armendáriz	garmy121t@yahoo.es / 0987382825	Scrum Team (Programador)
Milton Saltos	milton3552@hotmail.com / 0984752026	Scrum Team (Programador)

Elaborado por: Los Autores.

Diseño Conceptual

Recopilada la información necesaria para la elaboración de la propuesta, se realizó el diseño conceptual como se observa en la siguiente figura:

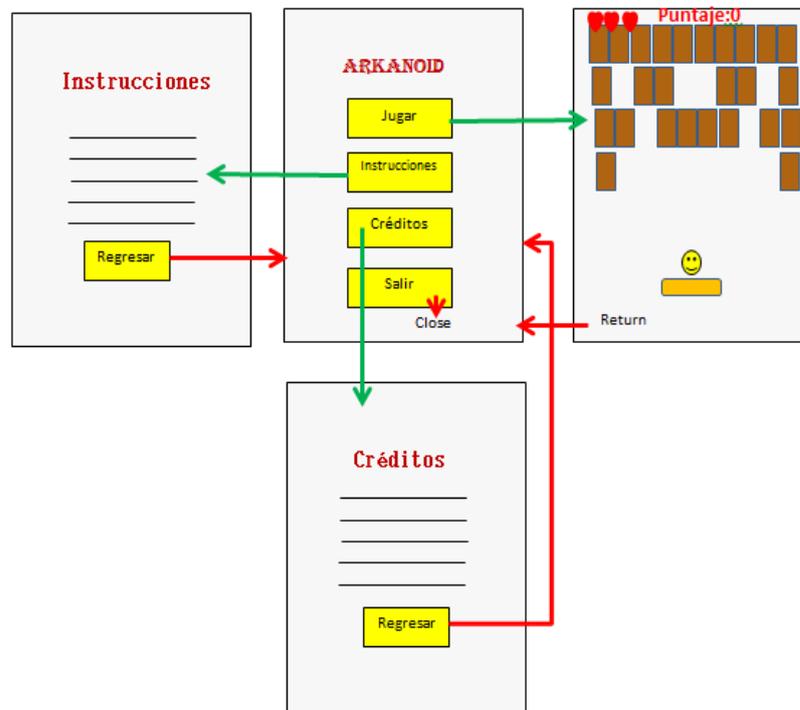


Figura III.24. Diseño Conceptual del videojuego.

Fuente: Los Autores.

Elementos del Videojuego

En el videojuego existirán los elementos que se observan en la **Tabla III.XI.**, cabe resaltar que las imágenes mostradas no son definitivas, sino tan solo bosquejos.

Tabla III.XI. Elementos del prototipo de videojuego en Scrum.

Imagen	Nombre	Descripción
	Corazón	Identifica los intentos o vidas que tiene el jugador.
	Bloque	Forman un esquema y desaparecen al ser tocados por la bola
	Bola	Rebota por toda la zona del juego tratando de eliminar los bloques.
	Pala	Impide que la bola salga de la zona de juego, haciéndola rebotar hacia los bloques para romperlos.

Elaborado por: Los Autores.

User Story

Los user stories que se muestran en la **Tabla III.XII.**, presentan las funcionalidades que tendrá el prototipo, son el resultado de escuchar al Product Owner.

Tabla III.XII. User Stories del prototipo de videojuego en Scrum.

Como	Quiero / Quiero que...	De modo que / Para
Jugador	Tener un menú principal	Pueda ingresar a jugar, visualizar las instrucciones, mostrar los créditos del videojuego y salir.
Jugador	Visualizar las instrucciones del juego	Saber cómo jugar.
Jugador	Visualizar los créditos del juego	Saber quién lo desarrolló.
Jugador.	La pala se encuentre en la parte inferior, y pueda moverse en forma horizontal	La bola no se caiga y rebotarla contra los bloques para romperlos.
Jugador	La bola se mueva por toda la pantalla, rebotando en cada lado	Pueda romper los bloques.

Jugador	Al romper los bloques con la bola se escuche un sonido de “golpe o rotura”.	Sea divertido y le de realismo.
Jugador	Ver mediante un gráfico las vidas	Conocer los intentos que tengo.
Jugador	Visualizar el puntaje de los bloques que voy rompiendo	Pueda conocer cuántos puntos voy acumulando.
Diseñador	La bola tenga el aspecto de un muñeco	Se muestre más divertida.
Diseñador	Los bloques formen un esquema	Se encuentren separados para que exista dificultad en el juego.
Diseñador	El jugador posea solo 3 vidas	Al agotarse las vidas se termine el juego y muestre al final un mensaje que indique que perdió el juego
Diseñador	Al romper todos los bloques finalice el juego	Indicar que ha terminado el juego y ha ganado.
Diseñador	Exista en la pantalla de créditos como en la de instrucciones un botón de regreso	Pueda volver al menú principal
Diseñador	Exista audio de fondo en la pantalla de menú como en el juego	Sea divertido.

Elaborado por: Los Autores.

Product Backlog

Para el desarrollo de este proyecto, hemos usado una referencia de 2 puntos por 8 horas para la estimación, es decir, una jornada completa.

Tabla III.XIII. Product Backlog del prototipo de videojuego en Scrum.

Id	Ítem	Prioridad	Estimación	Estado	Criterios de aceptación
US1	Como Jugador, quiero visualizar las instrucciones del juego, para saber cómo jugar.	2	1	Pendiente	# Debe existir en una pantalla todas las instrucciones del juego, en donde se indique como jugar.
US2	Como Jugador, quiero visualizar los créditos del juego,	2	1	Pendiente	# Debe existir en una pantalla la información de las personas que

	para saber quién lo desarrolló.				desarrollaron el juego, como sus nombres y apellidos.
US3	Como Jugador, quiero tener un menú principal, de modo que pueda ingresar a jugar, visualizar las instrucciones, mostrar los créditos del videojuego y salir.	2	2	Pendiente	# Debe existir un menú principal, en donde exista cuatro botones, que me permitirán, ingresar a jugar, ver las instrucciones del juego, mostrar los créditos, y salir de la aplicación
US4	Como Diseñador, quiero que exista en la pantalla de créditos como en la de instrucciones un botón de regreso, de modo que pueda volver al menú principal	2	1	Pendiente	# En las dos pantallas (instrucciones y créditos), debe existir un botón de regreso al menú principal.
US5	Como Diseñador, quiero que la bola tenga el aspecto de un muñeco, para que se muestre más divertida.	2	1	Pendiente	# La bola debe parecerse a un muñeco, debe poseer brazos y piernas, haciéndola ver divertida.
US6	Como Diseñador, quiero que los bloques formen un esquema, de modo que se encuentren separados para que exista dificultad en el juego.	1	2	Pendiente	# Todos los bloques deben formar un esquema en la pantalla, empezando desde la parte superior de la pantalla, hasta llegar al centro, dejando algunos espacios entre ellos.
US7	Como Jugador, quiero que la bola se mueva por toda la pantalla, rebotando en cada lado para que pueda romper los bloques.	1	2	Pendiente	# La bola debe empezar desde la parte inferior, sobre la pala, de empezar a moverse al dar clic en la pantalla, rebotar al toparse con los lados de la pantalla y al golpear los bloques, romperlos, es decir que desaparezcan.
US8	Como Jugador, quiero que al romper	2	1	Pendiente	# Al golpear la bola con los bloque debe

	los bloques con la bola, se escuche un sonido de “golpe o rotura”, para que sea divertido y le de realismo.				escucharse un sonido, simulando que se rompen los bloques
US9	Como jugador, quiero visualizar el puntaje de los bloques que voy rompiendo, de modo que pueda conocer cuántos puntos voy acumulando.	2	1	Pendiente	# En la parte superior y en el centro, debe aparecer el puntaje de los bloques que se van rompiendo, deben incrementarse de 100 en 100.
US10	Como Jugador, quiero que la pala se encuentre en la parte inferior, y pueda moverse en forma horizontal, de modo que la bola no se caiga y revotarla contra los bloques para romperlos.	1	2	Pendiente	# La pala debe estar en la parte inferior centrada y debe moverse en forma horizontal (ambos sentidos), presionando sobre ella y llevándola al lado que se desee.
US11	Como Diseñador, quiero que el jugador posea solo 3 vidas, de modo que al agotarse las vidas se termine el juego y muestre al final un mensaje que indique que perdió el juego.	1	2	Pendiente	# Al perder el jugador las tres vidas (intentos), el juego debe finalizar y mostrar un mensaje de “PERDISTE SIGUE PRACTICANDO..”.
US12	Como Jugador, quiero ver mediante un gráfico las vidas, para conocer los intentos que tengo.	2	1	Pendiente	# Mediante un gráfico de corazones rojos se debe mostrar en la parte superior izquierda de la pantalla el número de vidas o intentos que le queda al jugador.
US13	Como Diseñador, quiero que al romper todos los bloques	1	2	Pendiente	# Cuando ya se hayan roto todos los bloques y no exista ninguno, el

	finalice el juego, para indicar que ha terminado el juego y ha ganado.				juego debe finalizar y mostrarse en la pantalla un mensaje de "GANASTE".
US14	Como Diseñador, quiero que exista audio de fondo en la pantalla de menú como en el juego, para que sea divertido.	2	1	Pendiente	# Debe existir un mismo audio de fondo para las pantallas de menú principal, instrucciones, créditos y otro audio para el juego.

Elaborado por: Los Autores.

3.4.2. Fase Game

Sprints

Normalmente los sprints en Scrum son de 2 a 4 semanas, pero en el Sprint Planning hemos decidido hacerlo en una semana y todo el desarrollo del proyecto se ha dividido en 2 sprints.

Sprint 1

El objetivo del sprint 1 es obtener la primera versión del prototipo de videojuego "ARKANOID", para lograr este objetivo la duración de las reuniones diarias serán de 15 minutos y la fecha de revisión de la iteración será el día 11 de enero de 2013, este sprint implementará los user stories de la siguiente tabla:

Tabla III.XIV. User Stories a implementarse en el Primer Sprint.

User Story	Id	Tarea	Responsable	Tiempo Estimado
US6	T01	Formar un esquema en la pantalla con los bloques, separándolos distribuidamente.	Milton Saltos	8 horas
US7	T02	Dar movimiento a la bola por toda la pantalla, rebotando en cada lado y romper los bloques al colisionar.	Gabriel Armendáriz	8 horas
US10	T03	Ubicar a la pala en la parte inferior, de modo que pueda moverse en forma horizontal, para que la bola no se caiga y revotarla contra los bloques para romperlos.	Gabriel Armendáriz	8 horas

US11	T04	Darle al jugador solo 3 vidas, de modo que al agotarse se termine el juego y mostrar al final un mensaje que indique que perdió el juego.	Milton Saltos	8 horas
US13	T05	Finalizar el juego al romperse todos los bloques, he indicar que el juego a terminado y ha ganado.	Gabriel Armendáriz	8 horas
Total Horas				40 horas

Elaborado por: Los Autores.

En las reuniones diarias se actualizó las listas de tares del sprint Backlog.

En la **Figura III.25.**, se muestra la lista de tareas del Sprint 1 con las actividades realizadas al finalizar la primera iteración.

SPRINT		INICIO	DURACION					
1		07-ene-13	5	L	M	M	J	V
				07-ene	08-ene	09-ene	10-ene	11-ene
Tareas pendientes				5	3	2	0	
Horas de trabajo pendientes				40	24	12	0	
Requisito	Tarea	Responsable	Estado	ESFUERZO				
N/R	T01	Milton Saltos	Terminada	8				
T01	T02	Gabriel Armendáriz	Terminada	8				
T01,T02	T03	Gabriel Armendáriz	Terminada	8	8			
T03	T04	Milton Saltos	Terminada	8	8	4	0	
T03	T05	Gabriel Armendáriz	Terminada	8	8	8	0	

Figura III.25. Sprint Backlog del primer sprint.

Fuente: Los Autores.

En la **Figura III.26.**, se muestra el Burn Down en donde se indica cómo fue avanzando el proyecto a lo largo del sprint en relación a las tareas presentadas dentro de la iteración.



Figura III.26. Gráfico de Esfuerzo del primer Sprint.
Fuente: Los Autores.

La gráfica indica que la disminución de las horas fue constante de acuerdo a la pendiente que presenta; por lo que las tareas realizadas fueron manejadas y acopladas con éxito para la primera iteración del prototipo de videojuego.

La **Figura III.27.**, muestra el avance de las tareas realizadas a lo largo de la iteración en contra del tiempo hasta la fecha de finalización de la iteración.

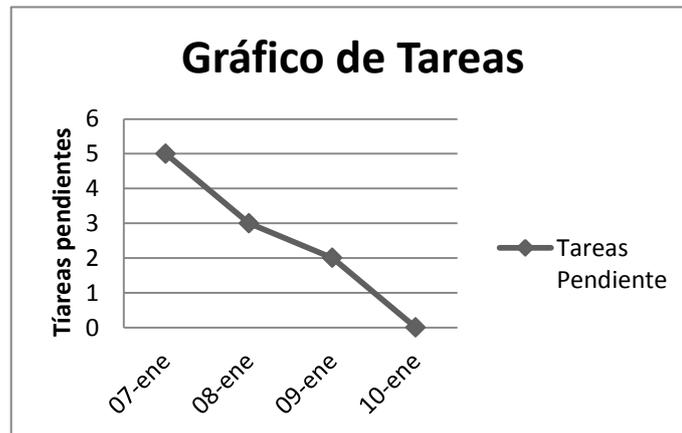


Figura III.27. Gráfico de Tareas del primer Sprint.
Fuente: Los Autores.

El gráfico permite analizar la distribución de las tareas, en la cual se observa que la evaluación de la estimación de las mismas en relación a la dificultad de completar las tareas fue buena.

Sprint Review Meeting

En esta reunión llevada a cabo el día 11 de enero de 2013, se presentó la funcionalidad del prototipo, es decir el sprint backlog 1 implementado, se realizó la demostración del videojuego en el entorno del cliente (jugador) como se muestra en la siguiente figura:



Figura III.28. Funcionalidad del prototipo al finalizar el Primer Sprint.

Fuente: Los Autores.

Conclusión del Sprint Review Meeting

Se cumplió con las metas establecidas, el trabajo estuvo bien distribuido con buenos tiempos.

Sprint Retrospective

En esta reunión se decidió seguir con las mismas prácticas, artefactos y herramientas utilizadas, debido a que fueron apropiadas para cumplir con éxito el primer sprint.

Después de realizar las reuniones de revisión y retrospectivo se dio por concluido el primer sprint.

Sprint 2

El objetivo del segundo y último sprint es obtener la versión final del prototipo de videojuego, de igual manera para lograr este objetivo la duración de las reuniones diarias seguirán siendo de 15 minutos y la fecha de revisión de la iteración será el día 18 de enero de 2013, este sprint implementará los User Stories faltantes, los mismos se muestran en la siguiente tabla:

Tabla III.XV. User Stories a implementarse en el Segundo Sprint.

User Story	Id	Tarea	Responsable	Tiempo Estimado
US1	T06	Implementar la pantalla de las instrucciones del juego.	Gabriel Armendáriz	4 horas
US2	T07	Implementar la pantalla de los créditos del juego.	Milton Saltos	4 horas
US3	T08	Implementar el menú principal, con 4 botones para que se pueda ingresar a jugar, visualizar las instrucciones, mostrar los créditos del videojuego y salir.	Gabriel Armendáriz	4 horas
US4	T09	Crear en la pantalla de créditos como en la de instrucciones un botón de regreso, para volver al menú principal.	Milton Saltos	4 horas
US5	T10	Diseñar a la bola con el aspecto de un muñeco, con brazos y piernas.	Gabriel Armendáriz	4 horas
US8	T11	Insertar un sonido de “golpe o rotura” al romper los bloques con la bola.	Milton Saltos	4 horas
US9	T12	Mostrar el puntaje de los bloques que son rotos con la bola.	Gabriel Armendáriz	4 horas
US12	T13	Mostrar mediante un gráfico las vidas o intentos del jugador.	Milton Saltos	4 horas
US14	T14	Insertar audio de fondo en la pantalla de menú y en el juego,	Gabriel Armendáriz	4 horas
Total Horas				36 horas

Elaborado por: Los Autores.

En la **Figura III.29.**, se muestra la lista de tareas del Sprint 2 con las actividades realizadas al finalizar la segunda y última iteración.

SPRINT		INICIO		DURACION							
2		14-ene-13		5							
						L	M	M	J	V	S
						14-ene	15-ene	16-ene	17-ene	18-ene	19-ene
Tareas pendientes						9	7	3	1	0	
Horas de trabajo pendientes						36	24	12	2	0	
Requisito	Tarea	Responsable	Estado	ESFUERZO							
N/R	T06	Gabriel Armendáriz	Terminada	4							
N/R	T07	Milton Saltos	Terminada	4							
T04,T05	T08	Gabriel Armendáriz	Terminada	4	2						
T06, T07	T09	Milton Saltos	Terminada	4	2						
N/R	T10	Gabriel Armendáriz	Terminada	4	4						
T03	T11	Milton Saltos	Terminada	4	4						
T03	T12	Gabriel Armendáriz	Terminada	4	4	4					
T05	T13	Milton Saltos	Terminada	4	4	4					
T08,T05	T14	Gabriel Armendáriz	Terminada	4	4	4	2	0			

Figura III.29. Sprint Backlog de la Segunda Iteración.

Fuente: Los Autores.

Como resultado de la segunda iteración se puede observar en la **Figura III.30.** y **III.31.**, las cuales corresponden al gráfico de esfuerzo del segundo y último sprint y el gráfico de las tareas realizadas en el tiempo, respectivamente.



Figura III.30. Gráfico de Esfuerzo del Segundo Sprint.

Fuente: Los Autores.

La gráfica anterior indica que la disminución de las horas fue constante de acuerdo a la pendiente que presenta, aunque al final hubo un pequeño cambio o inconveniente con una tarea, pero que no incidió en nada y se pudo terminar con éxito en el tiempo planificado.

Del mismo modo el siguiente gráfico nos permite analizar la distribución de las tareas, en el cual se observa que la evaluación de la estimación de las mismas en relación a la dificultad de completar las tareas fue buena.

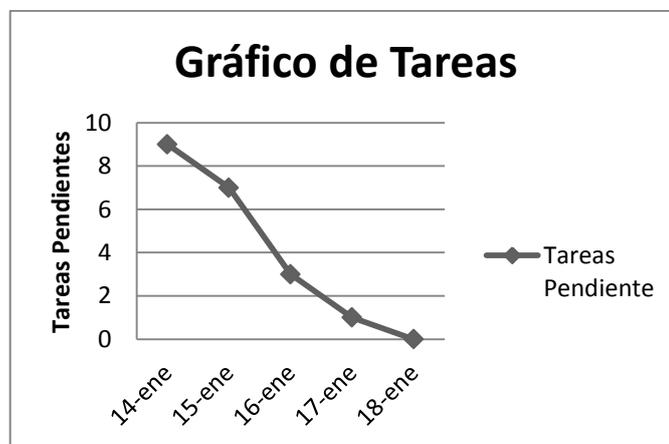


Figura III.31. Gráfico de Tareas del Segundo Sprint.
Fuente: Los Autores.

Sprint Review Meeting

En esta reunión llevada a cabo el día 18 de enero de 2013, se presentó la funcionalidad total del prototipo, es decir el sprint backlog 2 implementado e integrado con la funcionalidad obtenida en el primer sprint, igualmente se realizó la demostración del videojuego en el entorno del cliente (jugador) como se muestra en la siguiente figura:



Figura III.32. Funcionalidad del prototipo al finalizar el segundo y último sprint.
Fuente: Los Autores.

Conclusión del Sprint Review Meeting

Se cumplió con las metas establecidas, el trabajo estuvo bien distribuido con buenos tiempos, sin existir inconvenientes.

Sprint Retrospective

En esta reunión se concluyó que las prácticas, artefactos y herramientas utilizadas en todo el proyecto fueron apropiados para cumplir con éxito los sprints.

3.4.3. Fase Post-game

Completado con éxito todos los requerimientos funcionales del prototipo de videojuego, se da por concluido y se cierra el proyecto.

Se libera el prototipo de videojuego “ARKANOID” y se verifica la versión a entregar; no se generó la documentación final ni se realizó el pre-lanzamiento y el lanzamiento, debido a que para nuestro análisis no es necesario.

3.5. Prototipo “Metodología Ágil XGD”

Roles

Para el desarrollo del prototipo de videojuego, los roles fueron distribuidos entre los dos autores de esta tesis, como se observa en la siguiente tabla:

Tabla III.XVI. Roles asignados para el desarrollo del prototipo con XGD.

Persona	Contacto	Rol
Milton Saltos	milton3552@hotmail.com / 0984752026	Jefe del Proyecto
Gabriel Armendáriz	garmy121t@yahoo.es / 0987382825	Cliente
Milton Saltos	milton3552@hotmail.com / 0984752026	Development Team (Programador)
Gabriel Armendáriz	garmy121t@yahoo.es / 0987382825	Development Team (Programador)
Milton Saltos	milton3552@hotmail.com / 0984752026	Development Team (Diseñador)
Gabriel Armendáriz	garmy121t@yahoo.es / 0987382825	Development Team (Artista)
Gabriel Armendáriz	garmy121t@yahoo.es / 0987382825	Development Team (tester)

Elaborado por: Los Autores.

3.5.1. Fase Exploración

Primera Entrega

En esta fase se obtuvo los user stories para la primera entrega del videojuego, en donde el cliente plasmó mediante una lluvia de ideas, la información necesaria siendo esta: ubicación y movimiento de la pala, movimiento de la bola, esquema de bloques, vidas del jugador, finalización del juego, como se muestra en la siguiente figura:

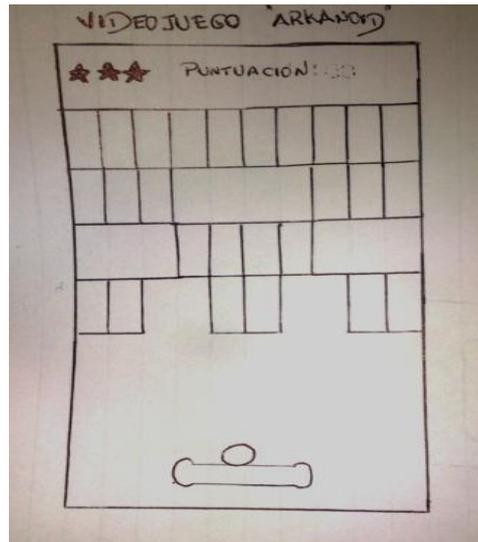


Figura III.33. Bosquejo obtenido para la primera entrega “XGD”.
Fuente: Los Autores.

User Stories Primera Entrega

Los user stories que se muestran en la **Tabla III.XVII.**, presentan la funcionalidad que tendrá el prototipo del videojuego para su primera entrega basadas en las ideas planteadas por el cliente.

Tabla III.XVII. User Stories Primera Entrega “XGD”.

USER STORIES PRIMERA ENTREGA
Los bloques formen un esquema con la finalidad de hallar dificultad
La bola se mueva por toda la pantalla rebotando y rompiendo los bloques
La pala se encuentre en la parte inferior, se mueva de forma horizontal y cuando la bola caiga esta rebote sobre la pala
Tener un máximo de 3 vidas y cuando pierdan todas el juego finalice
Cuando se rompan todos los bloques el juego finaliza y ha ganado.

Elaborado por: Los Autores.

Segunda Entrega

Igualmente en esta fase se obtuvo los user stories para la segunda y última entrega del videojuego, en donde el cliente planteó nuevas ideas funcionales para complementar el prototipo, siendo estas: menú principal, visualizar instrucciones y créditos, navegación entre pantallas, fondos y sonidos, como se muestra en la siguiente figura:

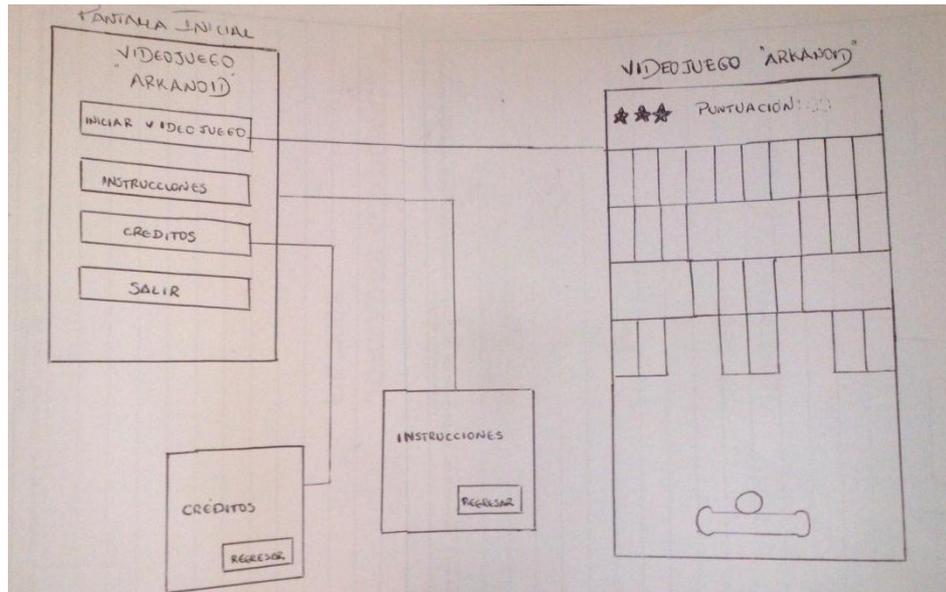


Figura III.34. Bosquejo obtenido para la segunda entrega “XGD”.

Fuente: Los Autores.

User Stories Segunda Entrega

Los user stories que se muestran en la **Tabla III.XVIII.**, presentan la nueva funcionalidad que tendrá el prototipo para su segunda y última entrega basadas en las ideas planteadas por el cliente.

Tabla III.XVIII. User Stories Segunda Entrega “XGD”.

USER STORIES SEGUNDA ENTREGA
Visualizar instrucciones del juego
Visualizar créditos del juego
Tener un menú principal
Navegación entre pantallas
La bola tenga un aspecto de balón de futbol.
Escuchar sonido cuando se rompan los bloques.
Visualizar el puntaje a medida que se van rompiendo los bloque.
Mejorar el aspecto de las vidas mediante la utilización de un gráfico.
Exista audio de fondo en la pantalla del menú principal.

Elaborado por: Los Autores.

3.5.2. Fase Planificación

Normalmente cada entrega en XGD tiene una duración de 2 a 3 semanas, pero en nuestro estudio hemos decidido ejecutarla en 1 semana.

Plannig Game

Primera entrega

Esta reunión tuvo una duración de 2 horas, se la realizó el día 21 de enero de 2013, en donde se estructuró los user stories, además se priorizaron y estimaron esfuerzos, tomando una referencia de 2 puntos por 8 horas para la estimación, es decir, una jornada completa, como se observa en la siguiente tabla:

Tabla III.XIX. Prioridades y Esfuerzo de los User Stories para la Primera Entrega

Id	User Stories	Prioridad	Esfuerzo
US01	Como Diseñador, quiero que los bloques formen un esquema, de modo que se encuentren separados para que exista dificultad en el juego.	1	8 horas
US02	Como Jugador, quiero que la bola se mueva por toda la pantalla, rebotando en cada lado para que pueda romper los bloques.	1	8 horas
US03	Como Jugador, quiero que la pala se encuentre en la parte inferior, y pueda moverse en forma horizontal, de modo que la bola no se caiga y revolarla contra los bloques para romperlos.	1	8 horas
US04	Como Diseñador, quiero que el jugador posea solo 3 vidas, de modo que al agotarse las vidas se termine el juego y muestre al final un mensaje que indique que perdió el juego.	1	8 horas
US05	Como Diseñador, quiero que al romper todos los bloques finalice el juego, para indicar que ha terminado el juego y ha ganado.	1	8 horas
Total Horas			40 horas

Elaborado por: Los Autores.

Segunda entrega

De igual manera esta reunión tuvo una duración de 2 horas, se la realizó el día 30 de enero de 2013, en donde se estructuró los últimos user stories para finalizar el prototipo, se dieron prioridades y se estimaron esfuerzos, siguiendo con la referencia de 2 puntos por 8 horas para la estimación, es decir, una jornada completa, como se observa en la siguiente tabla:

Tabla III.XX. Prioridades y Esfuerzo de los User Stories para la Segunda Entrega.

Id	User Stories	Prioridad cliente	Esfuerzo
US06	Como Jugador, quiero visualizar las instrucciones del juego, para saber cómo jugar.	2	4 horas
US07	Como Jugador, quiero visualizar los créditos del juego, para saber quién lo desarrolló.	2	4 horas
US08	Como Jugador, quiero tener un menú principal, de modo que pueda ingresar a jugar, visualizar las instrucciones, mostrar los créditos del videojuego y salir.	2	8 horas
US09	Como Diseñador, quiero que exista en la pantalla de créditos como en la de instrucciones un botón de regreso, de modo que pueda volver al menú principal	2	4 horas
US10	Como Diseñador, quiero que la bola tenga el aspecto de un balón de jugar futbol, para que se muestre más divertida.	2	4 horas
US11	Como Jugador, quiero que al romper los bloques con la bola, se escuche un sonido de “golpe o rotura”, para que sea divertido y le de realismo.	2	4 horas
US12	Como jugador, quiero visualizar el puntaje de los bloques que voy rompiendo, de modo que pueda conocer cuántos puntos voy acumulando.	2	4 horas
US13	Como Jugador, quiero ver mediante un gráfico las vidas, para conocer los intentos que tengo.	2	4 horas

US14	Como Diseñador, quiero que exista audio de fondo en la pantalla de menú como en el juego, para que sea divertido.	2	4 horas
Total Horas			40 horas

Elaborado por: Los Autores.

3.5.3. Fase Iteraciones

El prototipo de videojuego se ha dividido en 2 iteraciones, en la **Tabla III.XXI.**, se muestra las fechas de entrega para cada una de ellas.

Tabla III.XXI. Fechas de entrega de las iteraciones de XGD.

Numero Iteración	Fecha Inicio	Fecha Fin	Fecha Entrega
Iteración1	21 de Enero de 2013	28 de Enero de 2013	29 de Enero de 2013
Iteración2	30 de Enero de 2013	6 de Febrero de 2013	7 de Febrero de 2013

Elaborado por: Los Autores.

Reunión de Iteración

Esta reunión se la realizó al inicio de la iteración 1 y 2, el día 21 y 30 de enero de 2013 respectivamente, cada una tuvo una duración de 48 horas (2 días).

Para tener una mayor apreciación de los User Stories de la primera y segunda iteración, se realizó un detalle de cada uno de ellos. (Ver **Anexo I**)

Tabla III.XXII. Detalle User Story 01 “XGD”.

USER STORY 01	
Numero: US01	Nombre: Esquema de bloques
Usuario: Diseñador	
Numero de modificación del story: N/A	Iteración Asignada: Primera
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja

Descripción: Como Diseñador, quiero que los bloques formen un esquema, de modo que se encuentren separados para que exista dificultad en el juego.
Observaciones: Ninguna.

Elaborado por: Los Autores.

Para cada User Story se realizó la Prueba de Aceptación, que servirá para evaluar si cumplen con lo requerido. (Ver **Anexo I**)

Tabla III.XXIII. Prueba de Aceptación User Story 01 “XGD”.

Caso de Prueba de Aceptación 01	
Código: CP01	Historia de Usuario(Nro. y Nombre): US01- Diseñador
Nombre: Aspecto de Bola	
Descripción: Como Diseñador, quiero que los bloques formen un esquema, de modo que se encuentren separados para que exista dificultad en el juego.	
Condiciones de Ejecución: Los bloques deben formar un esquema con el propósito de que exista dificultad.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Juego. Los bloques aparecerán formando un esquema.	
Resultado Esperado: Funcionalidad correcta en la ejecución de todos los pasos.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Elaborado por: Los Autores.

También se realizó los Task Card para detallar de mejor manera los User Stories y asignar tareas a los programadores y diseñadores. (Ver **Anexo I**)

Tabla III.XXIV. Task Card del User Story 01-01 “XGD”.

Tarea de Ingeniería	
Número de Tarea: T01-01	Historias de Usuario (Nro. y Nombre): US01- Diseñador
Nombre Tarea: Imagen Bloque	
Tipo de Tarea: Diseño	Puntos Estimados: 3
Fecha Inicio: 23 de enero de 2013	Fecha Fin: 23 de enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: El bloque tiene un aspecto de edificio con el fin de dar mejor aspecto al juego	

Elaborado por: Los Autores.

Tabla III.XXV. Task Card del User Story 01-02 “XGD”.

Tarea de Ingeniería	
Número de Tarea: T01-02	Historias de Usuario (Nro. y Nombre): US01- Diseñador
Nombre Tarea: Esquema Bloques	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 23 de enero de 2013	Fecha Fin: 23 de enero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: Los bloques formaran un esquema para dar un poco de dificultad al juego.	

Elaborado por: Los Autores.

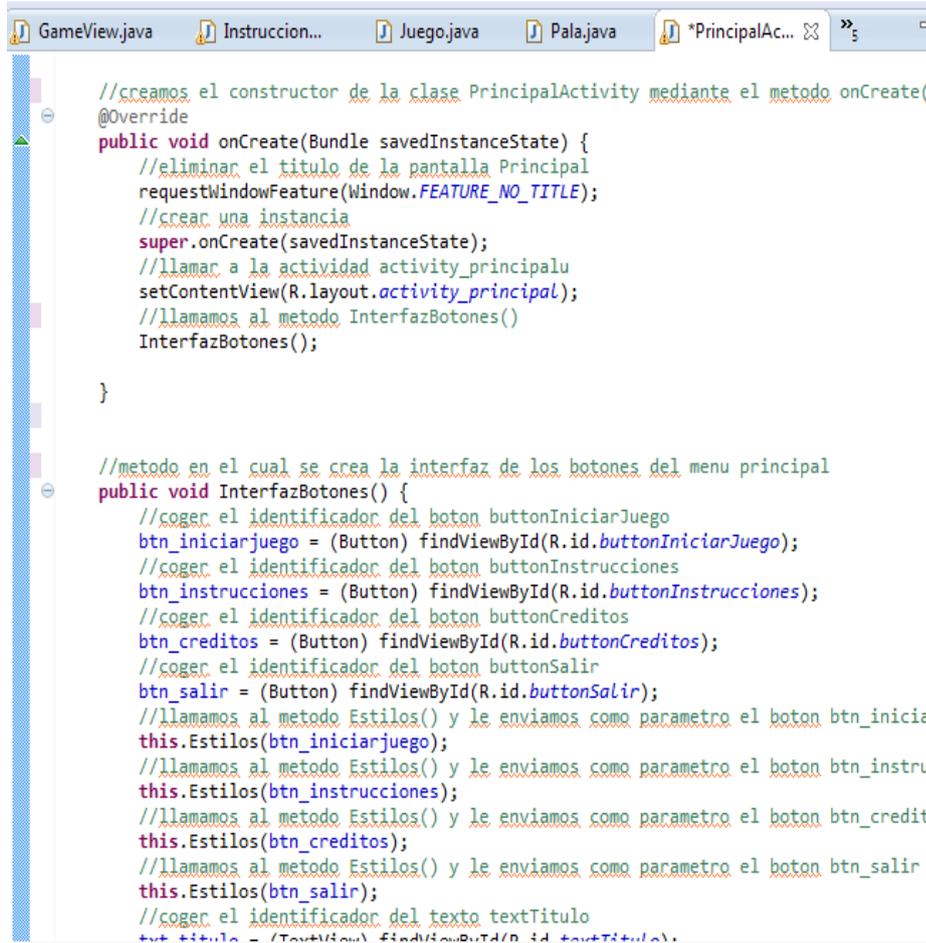
Una vez desarrolladas las tareas mediante los Card CRC se detallaron las clases, responsabilidades y colaboradores que formaron parte del desarrollo de cada User Story. (Ver **Anexo I**)

Tabla III.XXVI. Card CRC del User Story 01 “XGD”.

Nombre de la Clase: Bloques	
Responsabilidades: Las responsabilidades de esta clase son: Esquema.	Colaboradores: La Clase Bitmap
Atributos: private Bitmap bloque; private int[][] matriz;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Update() actualizar la bola cuando se está posicionando en la pantalla redimensionaBloques() es un método en el cual se da tamaño a los bloques, numero de filas y columnas.	

Elaborado por: Los Autores.

En todo el desarrollo del prototipo se realizó la refactorización de código con el IDE Eclipse como se muestra en la siguiente figura:



```
//creamos el constructor de la clase PrincipalActivity mediante el metodo onCreate(
@Override
public void onCreate(Bundle savedInstanceState) {
    //eliminar el titulo de la pantalla Principal
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    //crear una instancia
    super.onCreate(savedInstanceState);
    //llamar a la actividad activity_principalu
    setContentView(R.layout.activity_principal);
    //llamamos al metodo InterfazBotones()
    InterfazBotones();
}

//metodo en el cual se crea la interfaz de los botones del menu principal
public void InterfazBotones() {
    //coger el identificador del boton buttonIniciarJuego
    btn_iniciarjuego = (Button) findViewById(R.id.buttonIniciarJuego);
    //coger el identificador del boton buttonInstrucciones
    btn_instrucciones = (Button) findViewById(R.id.buttonInstrucciones);
    //coger el identificador del boton buttonCreditos
    btn_creditos = (Button) findViewById(R.id.buttonCreditos);
    //coger el identificador del boton buttonSalir
    btn_salir = (Button) findViewById(R.id.buttonSalir);
    //llamamos al metodo Estilos() y le enviamos como parametro el boton btn_inicia
    this.Estilos(btn_iniciarjuego);
    //llamamos al metodo Estilos() y le enviamos como parametro el boton btn_instru
    this.Estilos(btn_instrucciones);
    //llamamos al metodo Estilos() y le enviamos como parametro el boton btn_cred
    this.Estilos(btn_creditos);
    //llamamos al metodo Estilos() y le enviamos como parametro el boton btn_salir
    this.Estilos(btn_salir);
    //coger el identificador del texto textTitulo
    txt_titulo = (TextView) findViewById(R.id.txtTitulo);
```

Figura III.35. Refactorización del código.

Fuente: Los Autores.

No se realizó programación en parejas en ninguna iteración, debido a que solo somos 2 miembros de equipo.

En las 2 iteraciones se llevaron a cabo reuniones diarias que tuvieron una duración de 10 a 15 minutos en donde se trataron los inconvenientes que se encontraron al realizar las tareas correspondientes, además se hizo un seguimiento de las tareas de la primera y segunda iteración para comprobar el estado de cada una de ellas como se muestra en la **Figura III.36** y **Figura III.37** respectivamente, obtenidas al finalizar cada iteración.

TABLERO XGD PRIMERA ITERACIÓN FECHA: 28/01/2013

Id User Story	Id Tarea	Nombre Tarea	Estado	Responsable
US04	T01-01	Imagen Bloque	Finalizada	Milton Saltes

Figura III.36. Seguimiento del Estado de las Tareas de la Primera Iteración “XGD”.
Fuente: Los Autores.

TABLERO XGD SEGUNDA ITERACIÓN FECHA: 06/02/2013

Id USER STORY	Id TAREA	NOMBRE TAREA	ESTADO	RESPONSABLE
US06	T06-01	Pantalla Instrucciones	Finalizada	Milton Saltes
US07	T07-01	Pantalla Créditos	Finalizada	Gabriel Armentariz
US08	T08-01	Botones	Finalizada	Milton Saltes
US08	T08-02	Fondo Menú	Finalizada	Gabriel Armentariz
US09	T09-01	Salir pantalla créditos	Finalizada	Milton Saltes
US09	T09-02	Salir pantalla instrucciones	Finalizada	Gabriel Armentariz
US10	T10-01	Diseño Bola	Finalizada	Milton Saltes
US11	T11-01	Desaparecer Bloque	Finalizada	Gabriel Armentariz
US11	T11-02	Sonido de Impacto	Finalizada	Gabriel Armentariz
US12	T12-01	Puntaje	Finalizada	Milton Saltes
US13	T13-01	Perder Vidas	Finalizada	Milton Saltes
US14	T14-01	Fondo Musical	Finalizada	Gabriel Armentariz

Figura III.37. Seguimiento del Estado de las Tareas de la Segunda Iteración “XGD”.
Fuente: Los Autores.

No se finalizó en ninguna de las iteraciones en los tiempos establecidos, hubo un retraso de un día respectivamente, pero se desarrolló con éxito todos los User Stories planteados y al concluir cada iteración, se envió a producción todo el resultado funcional obtenido.

3.5.4. Fase Producción

En esta fase el Tester realizó las pruebas correspondientes de los user stories pertenecientes a la primera y segunda iteración del videojuego.

En la **Figura III.38.**, se muestra la parte funcional de la primera iteración obtenida el día 29 de enero de 2013.

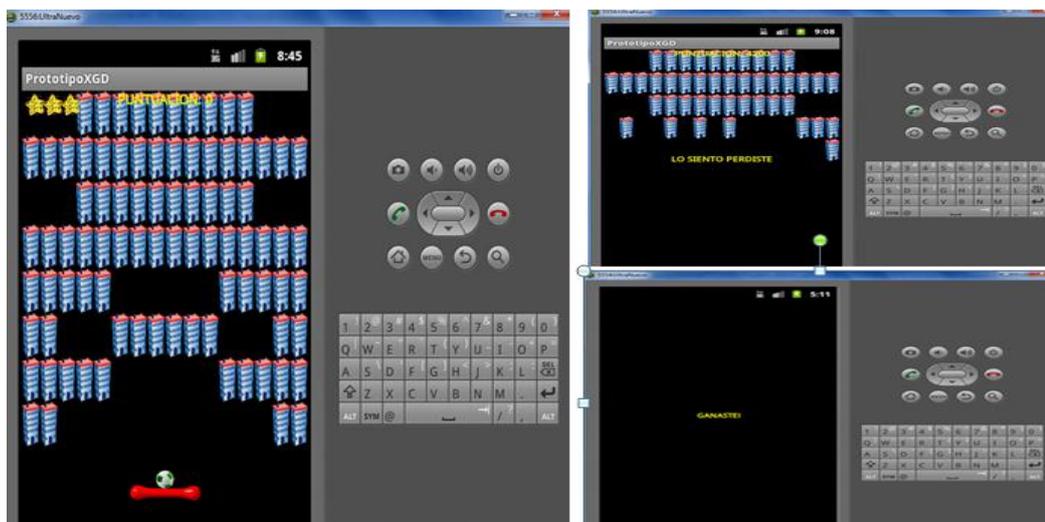


Figura III.38. Parte Funcional Primera Iteración “XGD”.
Fuente: Los Autores.

Se realizaron las pruebas de funcionamiento, obteniendo un resultado exitoso y cumpliendo con las pruebas de aceptación.

En la **Figura III.39.**, se muestra la parte funcional de la segunda y última iteración obtenida el día 06 de febrero de 2013.



Figura III.39. Parte Funcional Segunda Iteración “XGD”.
Fuente: Los Autores.

Se realizaron pruebas de funcionamiento de la segunda iteración integrada con el resultado obtenido en la primera iteración, dando un resultado exitoso, cumpliendo así con las pruebas de aceptación.

3.5.5. Fase Mantenimiento

La parte funcional de la primera iteración del prototipo de videojuego estuvo en funcionamiento mientras se encontraba en desarrollo la segunda y última iteración, no se realizaron cambios en ninguna entrega funcional.

3.5.6. Fase Muerte del Proyecto

Una vez concluido con todos los requerimientos funcionales del cliente y sin existir más user stories para desarrollar, se concluyó el proyecto y se entregó el prototipo de videojuego. No se realizó la documentación debido a que no es necesario para esta investigación.

3.6. Resultado del Análisis en cada Criterio

3.6.1. Planificación

Tabla III.XXVII. Pesos de los parámetros para las dos metodologías.

DESCRIPCIÓN	PESO	VALOR
PP1	EE	4
PP2	EE	4
PP3	EE	4
Total valoración		12 \cong 100%

Elaborado por: Los Autores.

Tabla III.XXVIII. Criterio “Planificación” Metodología Scrum.

Nº	PARÁMETRO	Scrum	Pi
1	Roles	Posee 3 roles principales: Product Owner, Scrum Master, Scrum Team y 2 roles auxiliares: Stakeholders, Managers.	4
2	Reuniones	Se realizan reuniones diarias (Daily Scrum, Scrum de Scrum), al iniciar cada sprint (Sprint Planning Meeting) y finalizar cada sprint (Sprint Review Meeting, Sprint Retrospective) permitiendo identificar y corregir problemas de inmediato.	4
3	Valores	Fundamenta el compromiso, enfoque, auto-organización, coraje, respeto y honestidad.	3

Elaborado por: Los Autores.

Tabla III.XXIX. Criterio “Planificación” Metodología XGD.

Nº	PARÁMETRO	XGD	Pi
1	Roles	Posee 7 roles principales: Development Team, Manager, Customer, Tester, Tracker, Coach, Gestor, y 2 roles auxiliares: Consultor, Doomsayer.	3
2	Reuniones	Se realizan reuniones diarias (Daily Stand Up Meeting), al planificar cada entrega (Planning Game) y al inicio de cada iteración (Iteration Meeting).	3
3	Valores	Fundamenta la simplicidad, retroalimentación, comunicación y coraje.	3

Elaborado por: Los Autores.

a. Interpretación de Resultados

$$\mathbf{Pascrum} = \Sigma (\mathbf{Pi})$$

$$\mathbf{Pascrum} = \Sigma (4 + 4 + 3)$$

$$\mathbf{Pascrum} = 11$$

$$\mathbf{Paxgd} = \Sigma (\mathbf{Pi})$$

$$\mathbf{Paxgd} = \Sigma (3 + 3 + 3)$$

$$\mathbf{Paxgd} = 09$$

$$\mathbf{Pc} = \Sigma (\mathbf{PPI})$$

$$\mathbf{Pc} = \Sigma (4 + 4 + 4)$$

$$\mathbf{Pc} = 12$$

$$\mathbf{Cscrum} = (\mathbf{Pascrum} / \mathbf{Pc}) * 100\%$$

$$\mathbf{Cscrum} = (11 / 12) * 100\%$$

$$\mathbf{Cscrum} = 91,67\%$$

$$\mathbf{Cxgd} = (\mathbf{Paxgd} / \mathbf{Pc}) * 100\%$$

$$\mathbf{Cxgd} = (09 / 12) * 100\%$$

$$\mathbf{Cxgd} = 75,00\%$$

Tabla III.XXX. Resultados del Criterio “Planificación”.

PARÁMETROS	METODOLOGÍAS					
	SCRUM			XGD		
	PPi	Pi	Cscrum	PPi	Pi	Cxgd
P1	4	4	91,67%	4	3	75,00%
P2	4	4		4	3	
P3	4	3		4	3	
TOTAL	12	11		12	09	

Elaborado por: Los Autores.

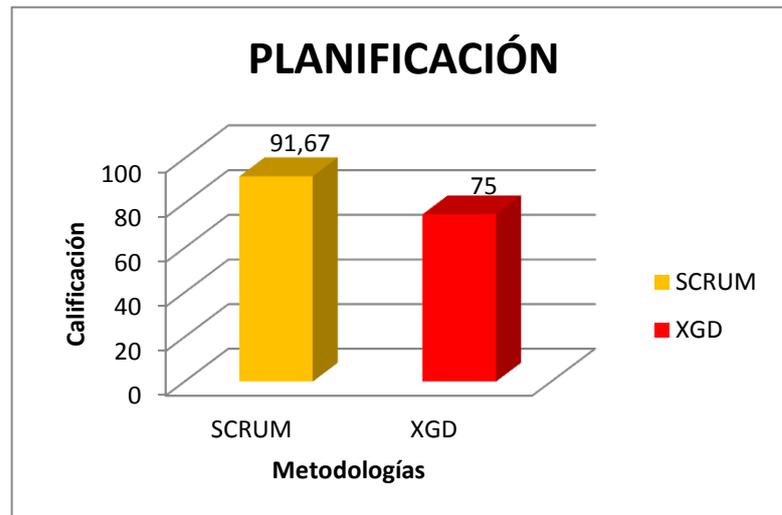


Figura III.40. Criterio “Planificación”.

Fuente: Los Autores.

b. Descripción de resultados del análisis

- Los roles de Scrum son óptimos para equipos de trabajo pequeños, concernidos y auto-organizados, donde la comunicación y la transparencia son totales, conteniendo así roles suficientes y específicos, evaluados y comprobados en el desarrollo del prototipo de videojuego, en el caso de XGD los roles son recomendables para equipos de trabajo grandes; además en esta metodología existen roles que pueden estar incluidos en un solo rol.

- Un pilar importante de Scrum son las reuniones, su importancia reside en que son la base para lograr transparencia y comunicación, se realizan a lo largo del proyecto, ya sea diariamente, al inicio de un sprint y al final de un sprint, de este modo se intercambia información, planifican sprints, discuten y solucionan problemas, evalúan resultados funcionales y se toman decisiones importantes para cumplir con éxito todo el proyecto, en el caso de XGD se realizan simplemente reuniones diarias y al inicio de cada iteración, resolviendo problemas y planificando las iteraciones en el desarrollo del proyecto.
- Los valores de XGD promueven el comportamiento individual, ligados al desarrollo del videojuego, mientras que los valores de Scrum promueven el comportamiento individual y grupal.
- Por todo lo expuesto anteriormente de acuerdo a los puntajes alcanzados para cada uno de los parámetros de evaluación se puede concluir que la metodología Scrum es la que brinda una mejor Planificación en el desarrollo de videojuegos alcanzado un porcentaje de 91,67% frente a la metodología XGD con 75%, por tal motivo Scrum y XGD se las califica como Excelente y Muy Buena respectivamente.

3.6.2. Ciclo de Vida del Proyecto

Tabla III.XXXI. Pesos de los parámetros para las dos metodologías.

DESCRIPCIÓN	PESO	VALOR
PP1	EE	4
PP2	EE	4
PP3	EE	4
PP4	EE	4
Total valoración		16 \cong 100%

Elaborado por: Los Autores.

Tabla III.XXXII. Criterio “Ciclo de vida del proyecto” Metodología Scrum.

Nº	PARÁMETRO	Scrum	Pi
1	Fases	Compuesta de tres fases: Pre-game, Game y Post-game, óptimas para desarrollo de videojuegos.	4
2	Actividades	Elaborar, Integrar, Revisar, Ajustar.	4
3	Prácticas	Centradas más en la forma de gestionar el proyecto.	3
4	Artefactos	Propone cinco artefactos: User Story, Product Backlog, Sprint Backlog, Burn-Down y Burn-Up, para mantener organizados los proyectos.	4

Elaborado por: Los Autores.

Tabla III.XXXIII. Criterio “Ciclo de vida del proyecto” Metodología XGD.

Nº	PARÁMETRO	XGD	Pi
1	Fases	Compuesta de seis fases: Exploración, Planificación, Iteración, Producción, Mantenimiento y Muerte del Proyecto.	2
2	Actividades	Escuchar, Diseñar, Codificar, Hacer Pruebas.	3
3	Prácticas	Equipo Completo, Diseño Incremental, Historias de Usuario, Ciclo Semanal, Integración Continua, Código Compartido, Reuniones Diarias de Seguimiento, El juego de la Planificación, Entregas Pequeñas, Pruebas, Refactorización, Desarrollo en Parejas	4
4	Artefactos	Propone cuatro artefactos: Tablero XGD, User Story (Prueba de Aceptación del User Story), Task Card, Card CRC.	3

Elaborado por: Los Autores.

a. Interpretación de Resultados

$$\mathbf{Pascrum} = \Sigma (\mathbf{Pi})$$

$$\mathbf{Pascrum} = \Sigma (4 + 4 + 3 + 4)$$

$$\mathbf{Pascrum} = 15$$

$$\mathbf{Paxgd} = \Sigma (\mathbf{Pi})$$

$$\mathbf{Paxgd} = \Sigma (2 + 3 + 4 + 3)$$

$$\mathbf{Paxgd} = 12$$

$$\mathbf{Pc} = \Sigma (\mathbf{PPi})$$

$$\mathbf{Pc} = \Sigma (4 + 4 + 4 + 4)$$

$$\mathbf{Pc} = 16$$

$$\mathbf{Cscrum} = (\mathbf{Pascrum} / \mathbf{Pc}) * 100\%$$

$$\mathbf{Cscrum} = (15 / 16) * 100\%$$

$$\mathbf{Cscrum} = 93,75\%$$

$$\mathbf{Cxgd} = (\mathbf{Paxgd} / \mathbf{Pc}) * 100\%$$

$$\mathbf{Cxgd} = (12 / 16) * 100\%$$

$$\mathbf{Cxgd} = 75,00\%$$

Tabla III.XXXIV. Resultados del Criterio “Ciclo de vida del proyecto”.

PARÁMETROS	METODOLOGÍAS					
	SCRUM			XGD		
	PPi	Pi	Cscrum	PPi	Pi	Cxgd
P1	4	4	93,75%	4	2	75,00%
P2	4	4		4	3	
P3	4	3		4	4	
P4	4	4		4	3	
TOTAL	16	15		16	12	

Elaborado por: Los Autores.

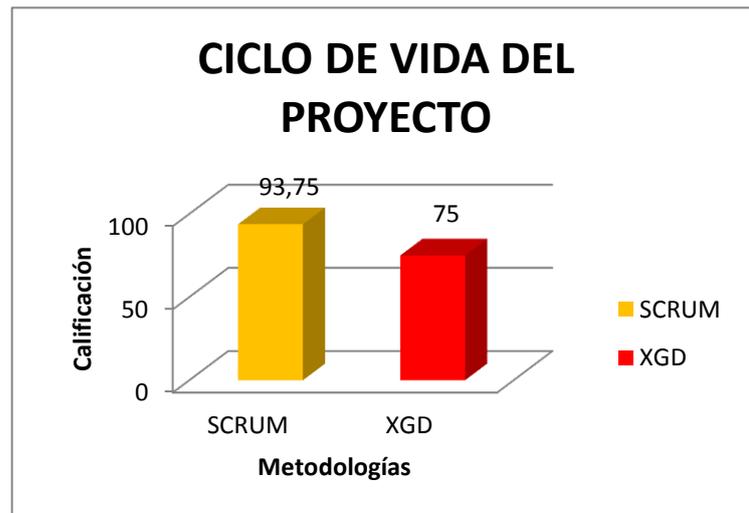


Figura III.41. Criterio “Ciclo de vida del proyecto”.
Fuente: Los Autores.

b. Descripción de resultados del análisis

- A través del análisis efectuado en la presente investigación, se determinó en base al desarrollo del prototipo de videojuego que las fases de Scrum son más eficientes que las de XGD, debido a que son las necesarias y suficientes para desarrollar un videojuego de calidad, fáciles a seguir, abarcan toda la gestión del proyecto y sobretodo cumplen con un seguimiento estricto en la planificación, tareas, sprints y tiempos de entregas establecidos, al contrario XGD cuenta con más fases, haciendo complejo el desarrollo del videojuego y aún más si son equipos sin experiencia en la utilización de la metodología.
- Scrum plantea actividades muy prácticas, fuertemente ligadas, ayudando a desarrollar de manera rápida y con una amplia visión de lo que se está haciendo, todas sus actividades son importantes, en cambio XGD parte con una pequeña visión al inicio fortaleciendo y complementando el desarrollo del videojuego según este vaya avanzando y siguiendo el orden establecido de las actividades planteadas, siendo la más importante la de Codificar.
- XGD provee prácticas de Ingeniería y Gestión del Proyecto, en cambio Scrum provee únicamente prácticas de Gestión del Proyecto.

- Los artefactos de Scrum, ayudan a planificar y revisar cada uno de los Sprints, aportando medios ineludibles para efectuar cada una de las reuniones, además nos permiten saber en cualquier momento qué tenemos que hacer, qué estamos haciendo y como estamos avanzando en base al tiempo estimado, de este modo se tiene un control integral del proyecto, en el caso de XGD permiten conocer a detalle los user stories proporcionando al equipo de desarrollo una mejor visión de lo que se quiere hacer, planifican las iteraciones pero no llevan un monitoreo de lo que se está realizando en base a los tiempos establecidos.
- Por todo lo expuesto anteriormente de acuerdo a los puntajes alcanzados para cada uno de los parámetros de evaluación se puede concluir que la metodología Scrum es la que posee un mejor Ciclo de Vida para el desarrollo de videojuegos alcanzado un porcentaje de 93,75% frente a la metodología XGD con 75%, por tal motivo Scrum y XGD se las califica como Excelente y Muy Buena respectivamente.

3.6.3. Calidad

Tabla III.XXXV. Pesos de los parámetros para las dos metodologías.

DESCRIPCIÓN	PESO	VALOR
PP1	EE	4
PP2	EE	4
PP3	EE	4
Total valoración		12 \cong 100%

Elaborado por: Los Autores.

Tabla III.XXXVI. Criterio “Calidad” Metodología Scrum.

N°	PARÁMETRO	Scrum	Pi
1	Fiabilidad	Trazabilidad entre los artefactos y simplicidad de los diseños.	4
2	Usabilidad	Evita la burocracia y la generación documental.	4
3	Mantenibilidad	No promueve la mantenibilidad.	0

Elaborado por: Los Autores.

Tabla III.XXXVII. Criterio “Calidad” Metodología XGD.

N°	PARÁMETRO	XGD	Pi
1	Fiabilidad	Simplicidad en el código y en los diseños.	4
2	Usabilidad	Evita la burocracia y la generación documental.	4
3	Mantenibilidad	Evitar mayor número de líneas y duplicación de código.	4

Elaborado por: Los Autores.

a. Interpretación de Resultados

$$\mathbf{Pascrum = \Sigma (Pi)}$$

$$Pascrum = \Sigma (4 + 4 + 0)$$

$$Pascrum = 08$$

$$\mathbf{Paxgd = \Sigma (Pi)}$$

$$Paxgd = \Sigma (4+4+4)$$

$$Paxgd = 12$$

$$\mathbf{Pc = \Sigma (PPi)}$$

$$Pc = \Sigma (4 + 4 + 4)$$

$$Pc = 12$$

$$C_{scrum} = (P_{scrum} / P_c) * 100\%$$

$$C_{scrum} = (08 / 12) * 100\%$$

$$C_{scrum} = 66,67\%$$

$$C_{xgd} = (P_{xgd} / P_c) * 100\%$$

$$C_{xgd} = (12 / 12) * 100\%$$

$$C_{xgd} = 100\%$$

Tabla III.XXXVIII. Resultados del Criterio “Calidad”.

PARÁMETROS	METODOLOGÍAS					
	SCRUM			XGD		
	PPi	Pi	Cscrum	PPi	Pi	Cxgd
P1	4	4	66,67%	4	4	100%
P2	4	4		4	4	
P3	4	0		4	4	
TOTAL	12	08		12	12	

Elaborado por: Los Autores.

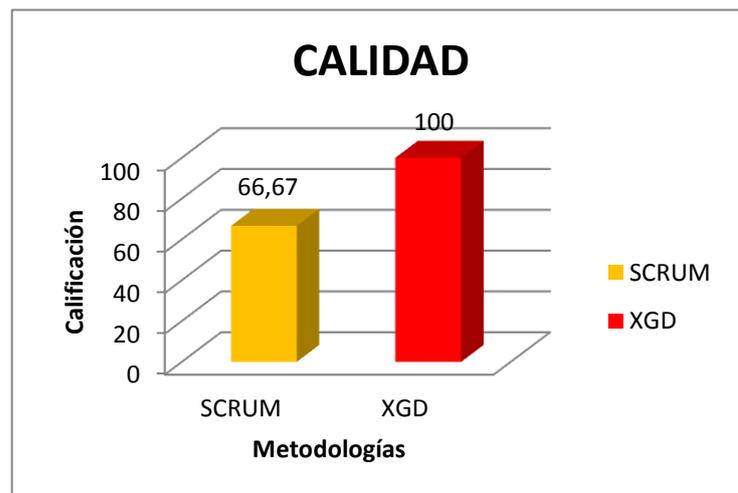


Figura III.42. Criterio “Calidad”.

Fuente: Los Autores.

b. Descripción de resultados del análisis

- En Scrum como en XGD existe trazabilidad entre sus artefactos debido a que ambas metodologías permiten identificar y registrar información referente a los user stories y a las tareas, desde su concepción hasta concluir el proyecto con suficiente información, permitiendo ver como se ha gestionado en cada fase exceptuado la última, además simplifican el diseño para agilizar el desarrollo y en el caso de XGD también la simplicidad se lleva a cabo a través de la refactorización constante para mejorar la calidad del código.
- Las dos metodologías no exigen documentar nada para iniciar un proyecto, lo más importante es que el videojuego funcione bien, cumpla con las necesidades, no hacen de la documentación un fin en sí mismo, sino indican que debe existir solo la documentación suficiente y necesaria, la misma que se realiza al finalizar el proyecto.
- XGD promueve mantenibilidad realizando refactorización de código conservándolo simple a medida que este va creciendo, cambiando su estructura sin modificar su comportamiento.
- Por todo lo expuesto anteriormente de acuerdo a los puntajes alcanzados para cada uno de los parámetros de evaluación se puede concluir que la metodología XGD es la que brinda una mejor Calidad en el desarrollo de videojuegos alcanzado un porcentaje de 100% frente a la metodología Scrum con 66,67%, por tal motivo XGD y Scrum se las califica como Excelente y Buena respectivamente.

3.6.4. Herramientas utilizadas en el desarrollo

Tabla III.XXXIX. Pesos de los parámetros para las dos metodologías.

DESCRIPCIÓN	PESO	VALOR
PP1	EE	4
PP2	EE	4
PP3	EE	4
Total valoración		12 \cong 100%

Elaborado por: Los Autores.

Tabla III.XL. Criterio “Herramientas utilizadas en el desarrollo” Metodología Scrum.

Nº	PARÁMETRO	Scrum	Pi
1	Gestión, medición y seguimiento	Herramientas pagadas: Scrum Timer, Kunagi, Pango Scrum, SpiraPlan, GreenHopper, entre otras, herramientas gratuitas: Scrum Desk, Banana Scrum, Sprintometer, Ice Scrum, Scrum Master Assistant, ScrumDo.	4
2	Refactorización	No existen herramientas que utilice la metodología en sí.	0
3	Integración	Herramienta pagada: Bamboo 4 y herramienta gratuita Hudson (Entornos Windows y Linux).	4

Elaborado por: Los Autores.

Tabla III.XLI. Criterio “Herramientas utilizadas en el desarrollo” Metodología XGD.

Nº	PARÁMETRO	XGD	Pi
1	Gestión, medición y seguimiento	Herramientas para la medición y rendimiento de aplicaciones: JMeter, herramientas de gestión de proyectos: Maven, Sprintometer y Ant.	4
2	Refactorización	Utiliza IDEs como Eclipse y NetBeans.	4
3	Integración	Herramienta gratuita: Cruise Control.	4

Elaborado por: Los Autores.

a. Interpretación de Resultados

Pascrum = Σ (Pi)

Pascrum = Σ (4 + 0 + 4)

Pascrum = 08

Paxgd = Σ (Pi)

Paxgd = Σ (4 + 4 + 4)

Paxgd = 12

Pc = Σ (PPi)

Pc = Σ (4+4+4)

Pc = 12

Cscrum = (Pascrum / Pc) * 100%

Cscrum = (08/12) * 100%

Cscrum r = 66,67%

Cxgd = (Paxgd / Pc) * 100%

Cxgd = (12/12) * 100%

Cxgd = 100%

Tabla III.XLII. Resultados del Criterio “Herramientas utilizadas en el desarrollo”.

PARÁMETROS	METODOLOGÍAS					
	SCRUM			XGD		
	PPi	Pi	Cscrum			Cxgd
P1	4	4	66,67%	4	4	100%
P2	4	0		4	4	
P3	4	4		3	4	
TOTAL	12	09		12	12	

Elaborado por: Los Autores.

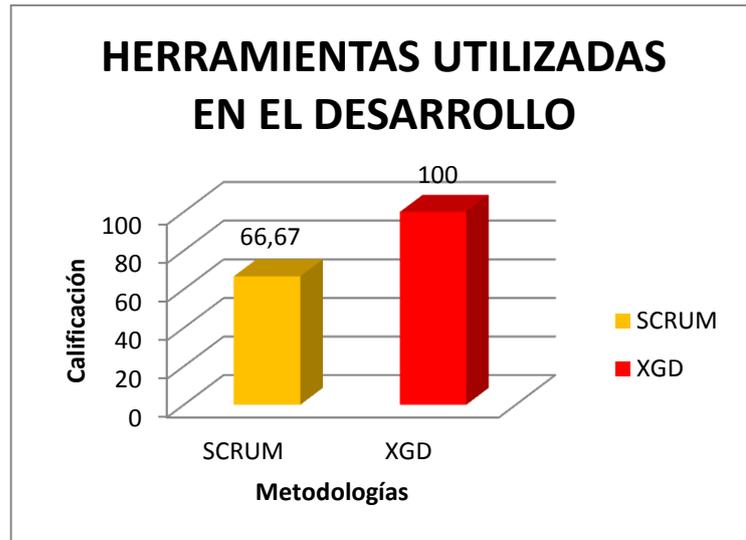


Figura III.43. Criterio “Herramientas utilizadas en el desarrollo”.
Fuente: Los Autores.

b. Descripción de resultados del análisis

- Se ha podido apreciar que hoy en día existen muchas herramientas excelentes con distribución libre o pagada que ayudan en el proceso de las dos metodologías ya sea para la gestión, medición, seguimiento, integración y refactorización, excepto esta última para Scrum, pero esto no quiere decir que no se pueda optar por una herramienta.
- Por lo expuesto anteriormente de acuerdo a los puntajes alcanzados para cada uno de los parámetros de evaluación se puede concluir que la metodología XGD por utilizar herramientas de refactorización ha alcanzado un porcentaje de 100% frente a la metodología Scrum con 66,67%, por no poseer herramientas de refactorización, por tal motivo XGD y Scrum se las califica como Excelente y Buena respectivamente.

3.7. Puntajes alcanzados

El puntaje final y el porcentaje obtenido en cada metodología se consiguen de la siguiente manera:

Puntaje Total del Análisis

$$PT = \sum (Pc)$$

Puntaje Total de Scrum

$$PT_{scrum} = ((\sum (P_{scrum}) / PT) * 100\%)$$

Puntaje Total de XGD

$$PT_{xgd} = ((\sum (P_{xgd}) / PT) * 100\%)$$

Tabla III.XLIII. Tabla General de Resultados.

CRITERIO	PARAMETRO	METODOLOGIAS		PESO
		SCRUM	XGD	PC
Planificación	Roles	4	3	4
	Reuniones	4	3	4
	Valores	3	3	4
Ciclo de vida del proyecto	Fases	4	2	4
	Actividades	4	3	4
	Prácticas	3	4	4
	Artefactos	4	3	4
Calidad	Fiabilidad	4	4	4
	Usabilidad	4	4	4
	Mantenibilidad	0	4	4
Herramientas utilizadas en el desarrollo	Gestión, medición y seguimiento	4	4	4
	Refactorización	0	4	4
	Integración	4	4	4
TOTAL		42	45	52

Elaborado por: Los Autores.

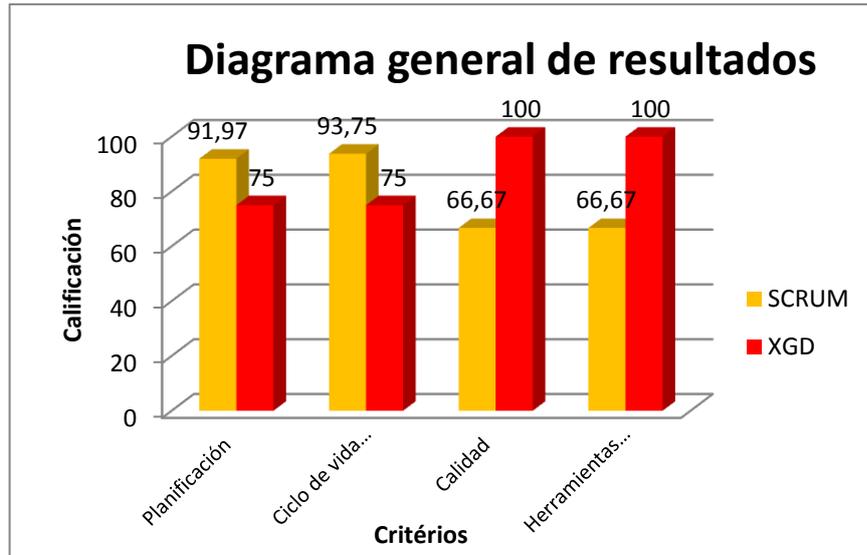


Figura III.44. Diagrama general de resultados.
Fuente: Los Autores.

Puntaje Total del Análisis

$$PT = \Sigma (Pc)$$

$$PT = 52$$

Puntaje Total de Scrum

$$PT_{scrum} = ((\Sigma (Pascrum) / PT) * 100\%)$$

$$PT_{scrum} = (42/52) * 100\%$$

$$PT_{scrum} = 80,77\%$$

Puntaje Total de XGD

$$PT_{xgd} = ((\Sigma (Paxgd) / PT) * 100\%)$$

$$PT_{xgd} = (45/52) * 100\%$$

$$PT_{xgd} = 86,53\%$$

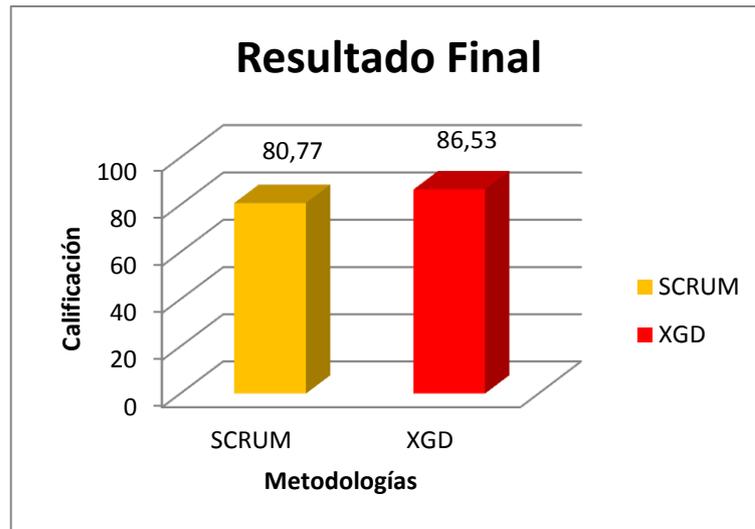


Figura III.45. Resultado final.
Fuente: Los Autores.

3.8. Resultado del Análisis

En base a todos los parámetros y criterios generales se ha comprobado que las dos metodologías se basan en el desarrollo iterativo e incremental, pero Scrum se enfoca en la gestión del proyecto, no posee prácticas de ingeniería, alcanzando un porcentaje final de 80,77%, a diferencia de XGD que es una metodología de ingeniería que contiene prácticas que aseguran la calidad focalizándose en el código, alcanzando un porcentaje final de 86,53%, calificando a las dos metodologías como Muy Buenas.

Por lo tanto, combinar SCRUM con XGD incrementa la productividad y la calidad del código desarrollado

Tabla III.XLIV. Calificación final de las Metodologías SCRUM y XGD.

DESCRIPCIÓN	CALIFICACIÓN
SCRUM	80,77%
XGD	86,53%

Elaborado por: Los Autores.

3.9. Ventajas y Desventajas

Mediante el análisis realizado podemos dar las siguientes ventajas y desventajas.

3.9.1. SCRUM

- **Ventajas**

- ✓ Mediante las reuniones frecuentes es más fácil comprobar el progreso del proyecto.
- ✓ Los miembros del equipo trabajan de manera más eficiente y con más calidad cuando ellos mismos se comprometen a entregar un resultado en un momento determinado y deciden cómo hacerlo, no cuando se les ha asignado una tarea e indicado el tiempo necesario para realizarla.
- ✓ Los miembros del equipo están más motivados cuando pueden usar su creatividad para resolver problemas y cuando pueden decidir organizar su trabajo.
- ✓ De manera regular el Product Owner redirige el proyecto en función de sus nuevas prioridades, de los cambios en el mercado, de los requisitos completados que le permiten entender mejor el producto.
- ✓ Al final de cada iteración el Product Owner puede aprovechar la parte del videojuego completada hasta ese momento para hacer pruebas con usuarios (jugadores) y tomar decisiones en función del resultado obtenido.
- ✓ Al ser un proceso iterativo incremental va mejorando el proceso en cada iteración.
- ✓ Desde la primera iteración el equipo tiene que gestionar los problemas que pueden aparecer en una entrega del proyecto. Al hacer patentes estos riesgos, es posible iniciar su mitigación de manera anticipada.
- ✓ Las reuniones se dedican a inconvenientes recientes, evitando el estancamiento.
- ✓ Buena respuesta a cambios.

- ✓ Al trabajar sobre una lista priorizada, nos aseguramos de que entregamos primero las características de mayor valor. Y al entregar un producto terminado al final de cada ciclo, no nos arriesgamos a que el proyecto fracase porque no haya dado tiempo a terminarlo.
- ✓ Al trabajar en ciclos de duración fija, se aprende rápidamente como de productivo es el equipo y cuántas tareas es capaz de terminar en un tiempo dado.
- ✓ En vez de trabajar de forma relajada al principio del proyecto y acumular estrés los últimos meses, con Scrum el estrés se distribuye al final de los múltiples ciclos de un mes que componen el desarrollo del producto.

- **Desventajas**

- ✓ No es óptima cuando los equipos de trabajos son muy grandes.
- ✓ Si los miembros del equipo no están centrados y convencidos, el proyecto nunca se completara o incluso fallará.
- ✓ Si una tarea no está bien definida, los costes de tiempo y dinero estimados del proyecto no serán demasiado exactos. En ese caso, la tarea se puede extender sobre varios sprints.
- ✓ Plantea un problema si el desarrollo está restringido por una fecha de entrega y un precio de entrega cerrados por contrato.
- ✓ Si algún miembro del equipo se marcha durante el desarrollo puede tener un efecto negativo enorme en el desarrollo del proyecto.

3.9.2. XGD

- **Ventajas**

- ✓ Existe una programación organizada utilizando refactorización de código mediante lo cual se reduce la tasa de errores y satisface al programador.
- ✓ Buenas prácticas en el desarrollo del videojuego.
- ✓ El cliente asigna y lleva un control sobre las prioridades de los user stories.
- ✓ Se hacen pruebas continuas durante el desarrollo del videojuego.
- ✓ Funciona muy bien cuando los requisitos son inciertos y volátiles
- ✓ Garantiza la satisfacción del cliente debido a que se encuentra involucrado en las constantes validaciones.

- **Desventajas**

- ✓ No es recomendable emplearlo en proyectos a corto plazo
- ✓ Altas comisiones en caso de fallar.
- ✓ Imposible prever todo antes de programar.
- ✓ Conseguir su implantación en un equipo es algo que puede resultar dificultoso.
- ✓ No es apropiada cuando el equipo de trabajo es menor de 6 personas.

3.10. Adaptación de las Metodologías

A continuación se realiza la adaptación de las metodologías ágiles Scrum y Extreme Game Development en una nueva metodología para desarrollo de videojuegos llamada DAV (Desarrollo Ágil de Videojuegos), en base al análisis anteriormente realizado.

DAV facilitará la creación de videojuegos de forma técnica y sencilla, la misma está formada por los siguientes elementos:

- Valores.
- Prácticas.
- Roles.

- Reuniones.
- Fases.
- Artefactos.
- Herramientas.

3.10.1. Valores

Se han seleccionado los valores tanto de Scrum como de XGD para complementarlos en la nueva metodología DAV, promoviendo el comportamiento individual y grupal de los miembros del equipo ligado al desarrollo del videojuego, a continuación se describe cada uno de ellos:

- **Compromiso:** Estar dispuesto para comprometerse a una meta. La metodología da a las personas la autoridad que necesitan para cumplir con sus compromisos.
- **Enfoque:** Enfocar todos los esfuerzos y habilidades para trabajar en lo que se comprometió hacer.
- **Auto-organización:** El Equipo DAV es auto-organizado con márgenes de decisión suficientes para tomar las decisiones que se consideren oportunas en las sucesivas Iteraciones.
- **Honestidad:** DAV mantiene todo acerca del proyecto visible a todos.
- **Respeto:** Los individuos están formados por sus orígenes y experiencias. Es importante respetar a las personas del equipo y sus diferentes formas de pensar.
- **Coraje:** Tener el coraje para comprometerse, actuar y esperar respeto.
- **Simplicidad:** Lo más importante es la simplicidad, no siempre se necesita que el videojuego sea innovador en todos los géneros, debe tener lo necesario.
- **Comunicación:** Se debe priorizar la comunicación de forma oral, de manera que las personas se sientan más cómodas para expresarse, no necesariamente con una gran cantidad de documentos cuya utilidad puede verse reducida.

- **Retroalimentación:** Está fuertemente ligada a la comunicación; la mejor manera de conocer el estado actual del videojuego es haciendo pruebas funcionales, esto proporcionará información real y confiable sobre el grado de fiabilidad.

3.10.2. Prácticas

Se han seleccionado de igual manera las prácticas tanto de Scrum como de XGD para integrar en la nueva metodología DAV, a continuación se describe cada una de ellas:

- **Equipo Completo:** El Equipo DAV debe permanecer en comunicación coherente, ser un todo.
- **Diseño Incremental:** Las tareas de un videojuego deben ser de la forma más sencilla posible, que funcionen correctamente.
- **Historias de Usuario:** Se trata de una breve descripción de las funcionalidades del videojuego, por lo general descrito por el Dueño del Producto.
- **Ciclos Semanales:** El proyecto está organizado y planificado para ser ejecutado en ciclos de corta duración.
- **Integración Continua:** En el proyecto existirán trozos de código fuente que continuamente se integrarán en funcionamiento.
- **Código Compartido:** Todo el Equipo DAV compartirá el código fuente del proyecto.
- **Reuniones Diarias de Seguimiento:** Reuniones rápidas del proyecto, con el objetivo de que todo el Equipo DAV esté consciente de la labor que se está realizando en un momento dado. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar.
- **Entregas Pequeñas:** Producir rápidamente versiones del videojuego que sean jugables, aunque no cuenten con toda la funcionalidad del videojuego. Esta versión ya constituirá un resultado de valor para el negocio.

- **Pruebas:** El videojuego será probado en varios momentos por diferentes personas, dando como resultado una información valiosa que podrá ser tomada como base para decisiones futuras.
- **Refactorización:** Es una actividad constante de reestructuración del código con el objetivo de evitar duplicaciones, mejorar la legibilidad, simplificarlo y hacerlo más flexible para facilitar posteriores cambios.
- **Desarrollo en Parejas (Cuando el Equipo DAV sea grande):** Toda la producción del videojuego debe realizarse con trabajo en parejas, pero simplemente los programadores y diseñadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño y mayor satisfacción del equipo).
- **Contacto Directo:** Los clientes se convierten en parte del equipo de desarrollo, de esta manera existe comunicación cara a cara.
- **Planes de Riesgos y Mitigación:** Se desarrollan planes de riesgos y mitigación frecuentes por parte del Equipo DAV, la mitigación de riesgos, la monitorización y la gestión de riesgos se lleva a cabo en todas las etapas y con compromiso.
- **Transparencia:** La transparencia en la planificación y desarrollo de módulos, permitirá a cada uno saber quién es responsable de qué y cuándo.
- **Frecuentes Reuniones:** Las frecuentes reuniones permite a las personas involucradas en el negocio monitorizar el progreso.
- **Planificación Ágil:** La planificación debe ser distribuida a lo largo del proyecto y no se concentrará en una determinada etapa. Las funciones más importantes serán planeadas con más detalle. Se debe evitar la planificación y estimación de las cosas que pueden no ser parte del videojuego.
- **Establecimiento de prioridades:** A lo largo del proyecto se llevan a cabo varias priorizaciones, que permiten al Equipo DAV trabajar siempre en lo que es más importante.

- **Estimación de Póquer:** Es una práctica ágil para reducir las dificultades habituales en las reuniones de trabajo para la planificación por juicio de expertos. En estas reuniones los participantes emplean un juego de cartas para concretar las unidades de esfuerzo que estiman para cada tarea.

3.10.3. Roles

Los roles de DAV se basan únicamente en los de la metodología Scrum, además se incluye en el Equipo DAV el rol de Tester, siendo estos necesarios y suficientes para ser aplicados en equipos pequeños, clasificándose en:

- Roles Principales.
- Roles Auxiliares.

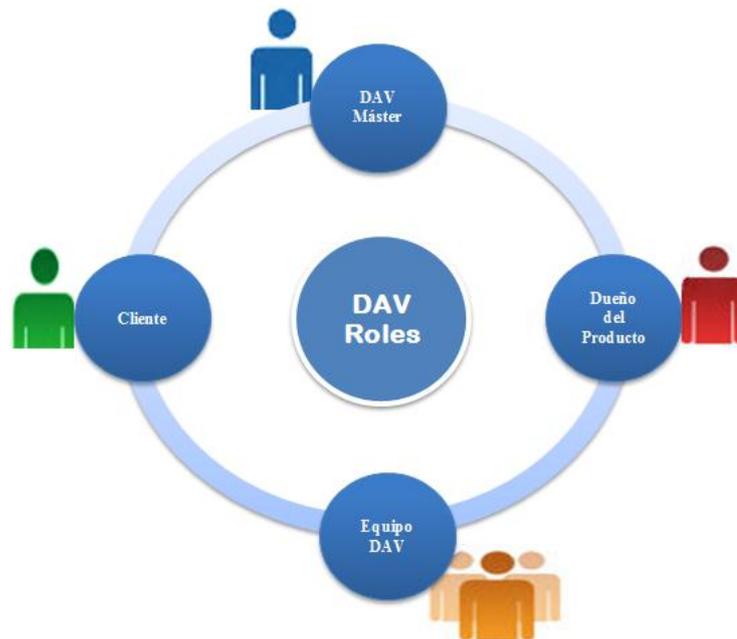


Figura III.46. Roles de DAV.
Fuente: Los Autores.

3.10.3.1. Roles Principales

Se define tres roles principales entre los cuales se dividen todas las responsabilidades del proyecto.

Dueño del Producto

El Dueño del Producto es la única persona autorizada para decidir sobre cuáles funcionalidades y características funcionales tendrá el videojuego. Es quien representa al cliente y todas aquellas partes interesadas en el videojuego, además es quien maneja y prioriza las funcionalidades a desarrollar y las coloca en la Reserva de Producto.

Funciones y responsabilidades:

- Encaminar las necesidades del videojuego, sabiendo escuchar a las partes interesadas y transmitir las al Equipo DAV.
- Revisar el videojuego e ir adaptándole funcionalidades, analizando las mejoras que éstas puedan otorgar un mayor valor.
- Ajuste de las características del videojuego.
- Responder por la rentabilidad del producto (ROI).
- Administración y priorización de la Reserva de Producto.
- Priorizar características de acuerdo al valor de mercado.
- Ajustar las características y prioridades cada cierto tiempo, según sea necesario.
- Aceptar o rechazar los resultados del desarrollo.

Aptitudes que debe tener:

- Excelente facilidad de comunicación en las relaciones interpersonales.
- Excelente conocimiento de videojuegos
- Facilidad para análisis de relaciones costo/beneficio
- Visión de negocios

DAV Máster

El DAV Máster no es un líder típico, sino es un auténtico servidor neutral, que será el encargado de enseñar la metodología DAV a cada integrante implicado en el proyecto, preocupándose de poner la metodología en práctica de modo que se encuentre dentro de la cultura de la organización y así entregue las ventajas previstas, asegurándose de que cada uno siga las reglas y prácticas de DAV.

Funciones y responsabilidades:

- Garantizar la correcta aplicación de DAV. Esto incluye, desde la correcta trasmisión de sus principios hasta la prevención de la inversión roles, es decir, guardar especial cuidado en que el Dueño del Producto no actúe en nombre del Equipo DAV y viceversa, o que la audiencia se inmiscuya en tareas que no le son propicias.
- Fomento del uso y respeto al proceso.
- Resolver los conflictos que dificulten el progreso del proyecto.
- Incentivar y motivar al Equipo DAV, creando un clima de trabajo colaborativo.
- Fomentar la auto-gestión del equipo e impedir la intervención de terceros en la gestión del equipo.
- Asegurar la calidad.
- Facilitar las reuniones de DAV de manera que sean productivas y consigan sus objetivos.

Aptitudes que debe tener:

- Excelentes conocimientos de DAV.
- Amplia vocación de servicio.
- Tendencia generosa.
- Amplia capacidad para la resolución de problemas.
- Analítico y observador.
- Saber incentivar y motivar.
- Capacidad docente e instructiva.
- Buen carisma para las negociaciones.

Equipo DAV

Es un equipo multidisciplinario y auto-organizado, su función principal es construir el videojuego que el Dueño del Producto especifica. Los miembros del equipo pueden ser de 3 a 9 personas, entre los cuales tenemos:

Diseñador

- Realiza el diseño conceptual.
- Define las reglas y la mecánica.
- Escribe el guión.

- Diseña los niveles.

Programador

- Implementa la funcionalidad pedida en el diseño.
- Programa la lógica del juego, gráficos (efectos), inteligencia artificial y audio.
- Conoce de herramientas/motores (infraestructura).

Artista

- Construye los objetos, personajes y niveles.
- Anima los personajes.

Sonidista

- Crea los sonidos, estilos, efectos y ambientes sonoros para el videojuego.

Tester

- Verifican que el contenido y funcionalidad del videojuego son correctos y están completos.
- Evalúan la jugabilidad.

Funciones y responsabilidades del Equipo DAV:

- Llevar la Reserva de Producto, a desarrollos potencialmente funcionales.
- Estimación del tamaño de los ítems de la Reserva de Producto.
- Compromiso de entregar incrementos del videojuego con calidad de producción.
- Seguimiento de su propio avance.
- Auto-organización, aunque con la responsabilidad ante el Dueño del Producto de entregar aquello que fue comprometido.

Aptitudes que deben tener:

- Ser profesionales expertos o avanzados en su disciplina.
- Tener vocación para trabajar en equipo.
- Capacidad de auto-gestión.

3.10.3.2. Roles Auxiliares

Los roles auxiliares son aquellos que no tienen un rol formal y no se involucran frecuentemente en el proceso DAV.

Interesados (Clientes)

Se refiere a la gente que hace posible el proyecto y para quienes el videojuego producirá el beneficio acordado que justifica su producción. Sólo participan directamente durante las revisiones de Iteración.

3.10.4. Reuniones

Las reuniones de DAV se basan únicamente en las de Scrum, siguen el mismo proceso, siendo estas un pilar muy importante.

3.10.4.1. Reunión Diaria de DAV

La reunión diaria facilita la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad. Dado que el equipo trabaja de manera colaborativa, este momento es esencial para asegurar un progreso continuo y evitar bloqueos; además el equipo medirá permanentemente su propio progreso en términos del objetivo de la Iteración.

Objetivos

- ✓ Mejorar la comunicación del Equipo DAV.
- ✓ Identificar y eliminar obstáculos.

Asistentes

- ✓ Únicamente el Equipo DAV.

Duración

- ✓ Máximo 15 minutos.

Entradas

- ✓ Reserva de Iteración, Gráfico Burn-Down y Burn-Up actualizados.

Salidas

- ✓ Reserva de Iteración, Gráfico Burn-Down y Burn-Up actualizados con datos de la reunión.
- ✓ Identificación de necesidades e impedimentos.

Proceso

- ✓ Uno a uno, los miembros del Equipo DAV explican brevemente:
 - a) ¿Qué has hecho desde ayer?
 - b) ¿Qué es lo que estás planeando hacer hoy?
 - c) ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?(Es el papel del DAV Máster recordar estos impedimentos)

3.10.4.2. Reunión de Planificación de la Iteración

Esta reunión se realiza al inicio de cada Iteración en donde se centra específicamente en auto-asignar tareas según sus intereses y se distribuye el trabajo a desarrollarse en la Iteración, deben estar presentes todos los miembros del Equipo DAV y ser ellos mismos quienes descompongan el trabajo en tareas y estimarlas.

Objetivo

- ✓ Determinar cuáles y cómo van a ser las funcionalidades que incorporará el videojuego en la Iteración.

Asistentes

- ✓ Equipo DAV, Dueño del Producto, DAV Máster, Interesados.

Duración

- ✓ Máximo 2 horas por cada semana que dure la Iteración.

Entradas

- ✓ Reserva del Producto priorizado.
- ✓ Producto realizado hasta el momento.
- ✓ Información adicional.

Salida

- ✓ Reserva de Iteración.
- ✓ Duración de la Iteración y fecha de entrega de la funcionalidad a obtener.
- ✓ Objetivo de la Iteración.

Proceso

En esta reunión encontramos dos partes:

1. Selección de las Historias de Usuario que entran en la Iteración.

Duración:

- Del 25% al 50% de la reunión.
- Debe decidirse al inicio de la reunión.

Actividades:

- El Dueño del Producto presenta con detalle las Historias de Usuario más prioritarias de la Reserva del Producto, mientras el Equipo DAV anota sus inquietudes y preguntas.
- El Equipo DAV pregunta al final de cada presentación todas las dudas que tenga.
- Todo dato que complemente las Historias de Usuario se deben anotar asociadas a éstas.
- Las aportaciones del Equipo DAV pueden modificar la priorización de la Reserva del Producto.
- La suma del esfuerzo en Puntos de Historia de las Historias de Usuario no debe superar la velocidad del Equipo.

2. Desglose en tareas de las Historias de Usuario seleccionadas.

Duración:

- Resto de la reunión.

Actividades:

- El Equipo DAV desglosa cada Historia de Usuario seleccionada para la Iteración en tareas, y estima el tiempo de resolución.
- Los miembros del Equipo DAV se auto-asignan las tareas (con una duración máxima de 8 horas), según intereses, conocimientos, curiosidad, etc.

3.10.4.3. Reunión de Revisión de la Iteración

Esta reunión se realiza al final de la Iteración, sirve para indicar al Dueño del Producto el resultado funcional del videojuego realizado por el Equipo DAV. Al mostrarse el incremento en funcionamiento, el Dueño del Producto y el equipo en general obtienen una retroalimentación que les permitirá evolucionar y darle un valor real a la Reserva del Producto.

Objetivo

- ✓ Presentar al Dueño del Producto y a todo aquel interesado el resultado de la Iteración.

Asistentes

- ✓ Todas las personas interesadas en la presentación.

Duración

- ✓ Máximo 1 hora.

Entradas

- ✓ Iteración terminada.

Salidas

- ✓ Feedback del Dueño del Producto para mejorar el videojuego.
- ✓ Convocatoria para la próxima reunión de planificación.

¿Qué no es la reunión de Revisión de la Iteración?

- ✓ No es una reunión de crítica de la Iteración.

- ✓ No es una reunión de crítica al Equipo DAV.

Proceso

1. Antes de la reunión, el Equipo DAV se auto-organiza para decidir cómo presentará las Historias de Usuario.
2. El miembro del Equipo DAV que presente el logro de la Iteración en su totalidad o parcialmente no será interrumpido, para darle fluidez a la demostración. Las preguntas, ruegos y sugerencias se realiza al final de la presentación.
3. Una vez presentadas todas las Historias de Usuario de la Iteración, se abre el tiempo de preguntas, sugerencias y debate en general.

Es importante tener en cuenta que de la funcionalidad presentada deben salir todas aquellas ideas y rectificaciones que alimenten la Reserva de Producto, por lo que es muy importante que ésta sea actualizada antes de la siguiente reunión de planificación.

3.10.4.4. Reunión de Retrospectiva de la Iteración

Esta reunión se realiza después de la revisión de la Iteración con todos los miembros del Equipo DAV, para analizar cómo ha ido avanzando la Iteración desde todos los puntos de vista y tratar de mejorarlos para garantizar el avance continuo de los procesos y desarrollos.

Objetivo

- ✓ Mejorar y evolucionar el proceso de gestión.

Asistentes

- ✓ Equipo DAV, Dueño del Producto, DAV Máster.

Duración

- ✓ Máximo 2 horas.

Entradas

- ✓ Lista de mejoras de reuniones de retrospectivas anteriores priorizada.

Salidas

- ✓ Lista priorizada de mejoras a aplicar, actualizada con datos de la reunión.

Proceso

1. Comentar las mejoras más prioritarias de la reunión de retrospectiva anterior en las que se ha trabajado y que se han aplicado, así como las que siguen apareciendo.
2. Cada miembro del Equipo DAV apuntará en una pizarra o papel qué cosas le han gustado de la última Iteración, obteniendo una lista de puntos positivos.
3. Cada miembro del Equipo DAV apuntará en una pizarra o papel qué cosas no le han gustado de la última Iteración, obteniendo una lista de puntos a mejorar.
4. Cada miembro del Equipo DAV tendrá tres votos para asignar entre los puntos a mejorar.
5. La lista de nuevos puntos a mejorar se reorganiza por número de votos conseguidos, y se incluirán en la lista de mejoras no trabajadas desde retrospectivas anteriores
6. Se escogerán un número finito de mejoras para aplicar durante la próxima Iteración (generalmente las que más votos ha conseguido, o las “X” primeras si hay votos similares).

3.10.5. Fases

La división de trabajo en el desarrollo de videojuegos no se distribuye por igual en sus iteraciones; para apoyar esta realidad DAV se basa únicamente en las fases de Scrum: pre-juego, juego y post-juego.

3.10.5.1. Fase Pre-juego

En esta fase se realiza la concepción de la idea del juego del Dueño del Producto con ayuda de los miembros del Equipo, es decir, los aspectos fundamentales que conformarán el videojuego:

- **Género:** Se debe especificar el género o géneros al que pertenece el videojuego para establecer las características básicas que tendrá en su posterior diseño.
- **Historia:** Se debe realizar un bosquejo de la trama o historia del videojuego a desarrollar, indicando qué se quiere contar y cómo se quiere contar.
- **Bocetos:** Se crean bocetos o diseños preliminares de los personajes y de dónde transcurrirá la acción del videojuego, ya sean decorados, ambientaciones, ropaje, música, movimientos, etc.
- **Aspecto:** A partir de los bocetos se define el aspecto gráfico y artístico del videojuego, colores, temas dominantes, musicalidad, técnicas de diseño 3D o 2D, posiciones de cámaras, etc.
- **Interfaz de Usuario:** Se define la manera de cómo interactuará el jugador con el videojuego y con qué mecanismos contará para ello.
- **Objetivos:** Cuáles son las metas del videojuego, de acuerdo a la historia de éste.
- **Reglas:** Qué cosas podemos hacer y cómo vamos a dejar que se hagan.
- **Características:** Especificación de las principales características de cada personaje del videojuego y de los elementos que intervienen en éste.
- **Jugabilidad:** Es aquí donde se definirá cómo se va a jugar, de qué manera se va a jugar, qué cosas podemos hacer en el juego y cómo va reaccionar el entorno del juego a las acciones del jugador a través del personaje. A su vez se debe establecer cómo será la curva de aprendizaje del jugador. Todo esto sin entrar en detalles gráficos, sonoros o de historia.

- **Diseño de Niveles:** Describir qué niveles, según la historia o dificultad, se tendrá, cómo serán éstos, cuántos serán, y qué dificultad y retos se plantearán en cada uno de ellos.
- **Requerimientos técnicos:** Establecer los requerimientos técnicos de equipo que necesitará el videojuego para poder ejecutarse.
- **Marketing:** Parte fundamental, debido a que muchos videojuegos de inversiones millonarias han ido al traste por una mala campaña de publicidad, para evitar esto hay que establecer las líneas de publicidad para el videojuego.

También se define la arquitectura, se realiza un diseño conceptual de la solución, se escoge el lenguaje de programación, plataforma y otras herramientas necesarias para el desarrollo.

Teniendo una idea clara de las funcionalidades del videojuego se realiza lo siguiente:

- * Desarrollo de la Reserva del Producto.
- * Determinación de la fecha de entrega y la funcionalidad de una o más versiones.
- * Selección de la versión más adecuada para desarrollo inmediato.
- * Selección del Equipo DAV para desarrollar las versiones.
- * Evaluación y control adecuado de los riesgos.
- * Estimación del costo de la versión, incluyendo desarrollo, material, marketing y formación.

Pre-juego reduce la necesidad de encontrar ese elemento elusivo de diversión durante la etapa de Juego y permite al equipo centrarse en la ejecución del videojuego, en lugar de experimentar con él.

3.10.5.2. Fase Juego

En esta fase se tiene bien definido el alcance del proyecto, por lo tanto ya existe una idea clara de lo que realmente es el videojuego y lo que se debe hacer.

Iteraciones

Esta fase que suele llamarse Iteración, consta de ciclos que pueden durar de 1 a 4 semanas (por lo general 1-2 semanas), donde se desarrollan las funcionalidades del videojuego, su duración debe ser constante en cada ciclo. Sin embargo, este requisito no es estricto debido a que a veces el tiempo de cada Iteración es diferente.

Una Iteración consta de las siguientes actividades:

- **Elaborar:** Se desarrollan las funcionalidades o requisitos del videojuego que se encuentran en la Reserva de Iteración.
- **Integrar:** Las funcionalidades desarrolladas son continuamente integradas en la nueva versión del videojuego.
- **Revisar:** Se examinan las funcionalidades desarrolladas en la Iteración para corregir posibles errores y probar que su funcionalidad sea correcta.
- **Ajustar:** Se adecua las funcionalidades del videojuego desarrolladas encajándolas correctamente, además se realiza la refactorización del código de las mismas.

Estas actividades no tienen una secuencia. A veces un elemento de la Reserva del Producto se tiene que desarrollar, integrar, y revisar cuando otros sólo debe ser revisado o ajustado.

En la siguiente figura se muestra el proceso de DAV.

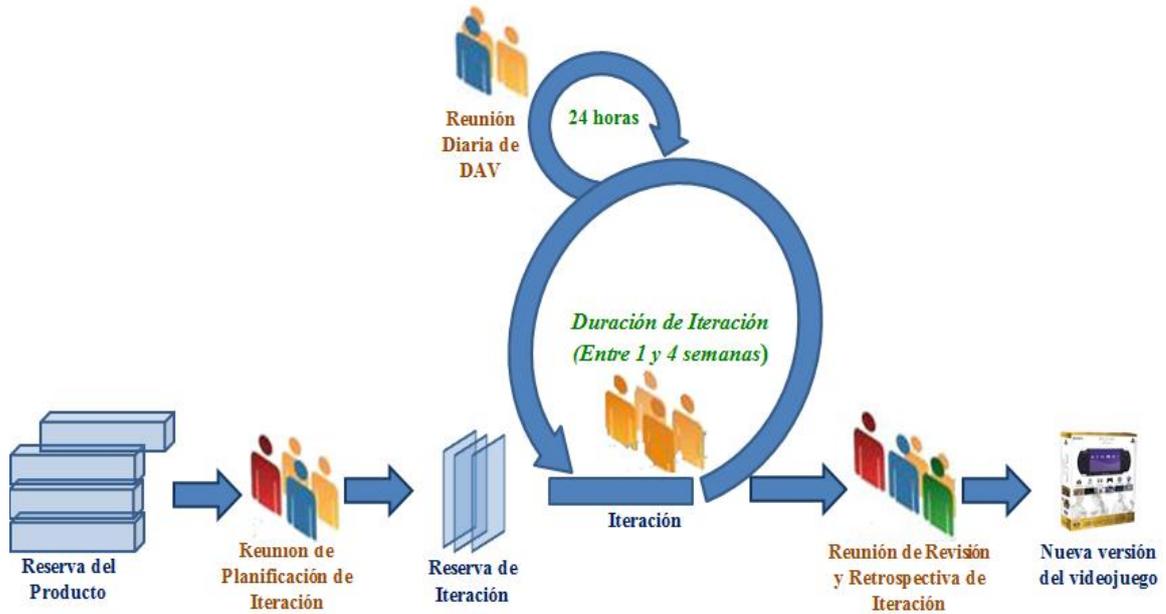


Figura III.47. Proceso de DAV.

Fuente: Los Autores.

Al comienzo de cada Iteración se realiza su planificación, donde se describen, priorizan y estiman las funcionalidades que se van a desarrollar, también se efectúan reuniones diarias para mejorar la comunicación del Equipo DAV y al concluir cada Iteración se evalúa su resultado.

3.10.5.3. Post-juego

Esta fase comienza cuando el equipo DAV ha desarrollado con éxito todas las Historias de Usuario y el Dueño del Producto ya no tiene más funcionalidades para implementar. Aquí se realiza el cierre del proyecto, donde se prepara la liberación del videojuego, se verifican las versiones a entregar, se genera la documentación final y se realiza el pre-lanzamiento y el lanzamiento.

3.10.6. Artefactos

En el caso de los artefactos, se han constituido en base a una selección mayoritaria de los artefactos de Scrum complementados con los de XGD, para mantener organizado el proyecto. Estos artefactos, ayudan a planificar y revisar cada uno de las Iteraciones,

aportando medios ineludibles para efectuar cada una de las reuniones, los mismos se detallan a continuación:

3.10.6.1. Historia de Usuario

Una Historia de Usuario es la definición de un requisito (normalmente una funcionalidad) que el Dueño del Producto quiere, con total claridad y sin ninguna ambigüedad, la cual el Equipo DAV deberá desarrollar como parte de una Iteración.

Son el resultado de escuchar al Dueño del Producto y ayudarle a resumir el requisito en una sola frase. Algo muy importante es que están escritas con el vocabulario del cliente, no con vocabulario técnico como se muestra en la siguiente figura:

ID	Como	Quiero.../ Quiero que....	De modo que... / Para...
HU01	Jugador	Visualizar las instrucciones del juego	Saber cómo jugar
HU02	Jugador	La pala se encuentre en la parte inferior y pueda moverse en forma horizontal	La bola no se caiga y revotarla contar los bloques para romperlo
HU03	Jugador	Al romper los bloques con la bola se escuche un sonido de "golpe o ruptura"	Sea divertido y le de realismo
HU04	Jugador	Ver mediante un gráfico las vidas	Conocer los intentos que tengo
HU05	Jugador	Visualizar el puntaje de los bloques que voy rompiendo	Pueda conocer cuántos puntos voy acumulando
HU06	Diseñador	La bola tenga el aspecto de un muñeco	Se muestre más divertida

Figura III.48. Historia de Usuario de DAV

Fuente: Los Autores.

Cada Historia de Usuario provoca una serie de preguntas acerca de los múltiples contextos en que se puede dar, por ejemplo:

- ♣ ¿Qué hace el videojuego si la pelota colisiona con un bloque?
- ♣ ¿Qué sucede si se ha agotado las vidas del jugador?

Las respuestas a estas preguntas son afirmaciones, ejemplos, los cuales se deben transformar en Pruebas de Aceptación. Por tanto, cada Historia de Usuario tiene asociados una o varias Pruebas de Aceptación.

Características

Las Historias de Usuario deben ser:

- ✓ Independientes unas de otras: De ser necesario, combinar las historias dependientes o buscar otra forma de dividir las historias de manera que resulten independientes.
- ✓ Negociables: La historia en sí misma no es lo suficientemente explícita como para considerarse un contrato, la discusión con el Dueño del Producto debe permitir esclarecer su alcance.
- ✓ Valoradas por el Dueño del Producto: Cada historia debe ser importante para el Dueño del Producto que para el Equipo DAV.
- ✓ Estimables: Un resultado de la discusión de una Historia de Usuario es la estimación del tiempo que tomará completarla. Esto permite estimar el tiempo total del proyecto.
- ✓ Pequeñas: Las historias muy largas son difíciles de estimar e imponen restricciones sobre la planificación de un desarrollo iterativo. Generalmente se recomienda la consolidación de historias muy cortas en una sola historia.
- ✓ Verificables: Las Historias de Usuario cubren requerimientos funcionales, por lo que generalmente son verificables en cada entrega.

3.10.6.2. Prueba de Aceptación

Las Pruebas de Aceptación validan el grado de satisfacción del Dueño del Producto, es decir confirman que una Historia de Usuario ha sido implementada correctamente.

Este tipo de pruebas son comúnmente realizadas por el Dueño del Producto y supervisadas por el Tester, informando de todas las deficiencias o errores que se

encuentre antes de dar por aprobado el requisito definitivamente. En la siguiente figura se muestra el formato de la Prueba de Aceptación.

Prueba de Aceptación	
<i>Identificador: PA001</i>	<i>Historia de Usuario (Nro. Y Nombre): HU001- Visualizar las Instrucciones del videojuego</i>
<i>Nombre: Visualizar las instrucciones del videojuego</i>	
<i>Descripción: Visualizar en una ventana visualizar las instrucciones del videojuego, además mediante un botón de fondo amarillo debe permitir regresar al menú principal.</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil, además debe estar terminado el menú inicio.</i>	
<i>Entrada / Pasos de ejecución:</i> <ul style="list-style-type: none">○ Escoger la opción “Instrucciones” en el Menú Principal.○ En la pantalla que aparece las instrucciones del videojuego, para regresar al Menú Principal dar clic en el botón Regresar.	
<i>Resultado Esperado: Mostrar las instrucciones del videojuego en una ventana, además tener un botón que permita salir de la ventana y regresar al menú principal.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Figura III.49. Prueba de Aceptación de DAV.
Fuente: Los Autores.

Dónde:

- **Identificador:** Es un identificador único de la Prueba de Aceptación para futuras referencias.
- **Historia de Usuario (Nro. Y Nombre):** El identificador y nombre de la Historia de Usuario a la cual pertenece la Prueba de Aceptación.
- **Nombre:** La Prueba de Aceptación debe tener un nombre entendible para cualquier persona, facilitando la comprensión tanto del propósito de la prueba como del campo de aplicación.
- **Descripción:** Contiene una breve descripción del propósito de la prueba y la funcionalidad que examina.

- **Condiciones de Ejecución:** Contiene información acerca de hardware o software necesario para poder ejecutar la prueba.
- **Entrada / Pasos de Ejecución:** Pasos a realizar para completar la prueba.
- **Resultado Esperado:** Contiene una descripción de lo que el analista debería ver tras haber completado todos los pasos de la prueba.
- **Evaluación de la Prueba:** Indica el resultado cualitativo de la ejecución de la prueba, a menudo con un Correcto/Fallido.

Características

- ✓ Aseguran que la funcionalidad del videojuego se cumpla satisfactoriamente.
- ✓ Marcan el camino a seguir, indicándole al Equipo DAV las funcionalidades más relevantes.
- ✓ Permiten al Dueño del Producto saber cuándo el proyecto está funcionando.
- ✓ Permiten al Equipo DAV conocer qué es lo que resta por hacer.
- ✓ Tienen una importancia crítica para el éxito de una iteración
- ✓ Deben estar escritas lo antes posible a partir del comienzo de la iteración.
- ✓ Sirve para que todo el equipo conozca los resultados de la Iteración
- ✓ Permiten evaluar el desempeño del Equipo DAV.
- ✓ Indican cuando una funcionalidad de la Iteración ha sido completada exitosamente.

3.10.6.3. Reserva del Producto

La Reserva del Producto es un listado de alto nivel, dinámico y públicamente visible para todos los involucrados en el proyecto.

En ella, el Dueño de Producto, mantiene una lista actualizada de requerimientos funcionales para el videojuego. Esta lista, representa “qué es lo que se pretende hacer”

pero sin mencionar “cómo hacerlo”, debido a que esta última, será tarea del Equipo DAV.

La Reserva del Producto es creada y modificada únicamente por el Dueño de Producto. Durante la reunión de planificación, el Equipo DAV obtendrá los ítems del videojuego, que deberá desarrollar durante la Iteración. En la siguiente figura se muestra el formato de la Reserva del Producto.

Prioridad	Nro. Historia de Usuario	Ítem	Estimación	Estado	Prueba de Aceptación
1	HU01	Como Jugador, quiero visualizar las instrucciones del juego, para saber cómo jugar	3	En curso	CA01
2	HU02	Como jugador, quiero visualizar el puntaje de los bloques que voy rompiendo, de modo que pueda conocer cuántos puntos voy acumulando.	5	Pendiente	CA02

Figura III.50. Reserva del Producto.

Fuente: Los Autores.

Para cada ítem, es necesario especificar:

El grado de prioridad

Los ítems de la Reserva del Producto, deben guardar un orden de prioridad, cuya base se apoye en:

- Beneficios de implementar una funcionalidad.
- Pérdida o costo que demande posponer la implementación de una funcionalidad.
- Riesgos de implementarla.
- Coherencia con los intereses del negocio

Esfuerzo que demanda

DAV, propone la estimación de esfuerzo y complejidad que demanda el desarrollo de las funcionalidades, solo para aquellas cuyo orden sea prioritario.

Estas estimaciones, no se efectúan sobre ítems poco prioritarios ni tampoco sobre aquellos donde exista un alto grado de incertidumbre.

De esta manera, se evita la pérdida de tiempo en estimaciones irrelevantes, postergándolas para el momento en el cual realmente sea necesario comenzar a desarrollarlas.

Prueba de Aceptación

Es necesario especificar para cada ítem de la Reserva del Producto, la o las Pruebas de Aceptación que debe superar, para considerar cumplido el requisito.

Ventajas de construir la Reserva del Producto

- ✓ Permite tener una perspectiva de todo lo que se quiere hacer y tener claras las prioridades del cliente.
- ✓ Ayuda a que el Equipo DAV sea más auto-disciplinado y respete las prioridades del Dueño del Producto.
- ✓ Permite que el Dueño del Producto pueda introducir cambios durante la vida del proyecto.
- ✓ Ayuda a manejar la incertidumbre durante el proyecto porque induce a describir con más detalle las Historias de Usuario más importantes y a relativizar la importancia de detallar historias de menor prioridad.
- ✓ Más ligero que un documento de requisitos exhaustivo.
- ✓ Asegura al Equipo DAV de entregar el trabajo que maximiza el beneficio comercial para el Dueño del Producto.

3.10.6.4. Tarjeta de Tarea

Las Tarjetas de Tarea se usan para describir las tareas que se realizarán en el proyecto y representar las responsabilidades asignadas a cada miembro del Equipo DAV. Estas tareas pueden ser de desarrollo, corrección o mejora, contienen número y nombre de la tarea, Historia de Usuario a desarrollar, fecha de inicio y fin de la tarea, se nombra al miembro del equipo responsable y presenta una descripción de la tarea como se muestra en la siguiente figura.

Tarea de Ingeniería	
<i>Número de Tarea:</i>	<i>Historia de Usuario (Nro. Y Nombre):</i>
<i>Nombre Tarea:</i>	
<i>Tipo de Tarea:</i> <i>Desarrollo / Corrección / mejora / otros</i>	<i>Puntos Estimados:</i>
<i>Fecha Inicio:</i>	<i>Fecha Fin:</i>
<i>Miembro Responsable:</i>	
<i>Descripción:</i>	

Figura III.51. Tarea de Ingeniería DAV.

Fuente: Los Autores.

3.10.6.5. Reserva de Iteración

La Reserva de Iteración es un cuadro altamente visible, donde se define el trabajo o las tareas que el Equipo DAV desarrollará para poder convertir la Reserva del Producto seleccionado en una Iteración, en un incremento potencialmente funcional del videojuego.

Generalmente la Reserva del Producto se visualiza mediante tableros físicos que hacen visible el proceso de construcción a toda persona que ingrese al área de desarrollo.

Para armar la Reserva del Producto, el Equipo DAV divide los ítems en tareas, de modo que cada una se desarrolle en un lapso de entre 4 a 16 horas, si una tarea es mayor a 16 horas, deberá ser dividida en otras menores. Las tareas en la Reserva de Iteración nunca

son asignadas, son tomadas por los miembros del Equipo DAV de modo que les parezca oportuno y solamente ellos pueden cambiar la Reserva de Iteración, en la siguiente figura se muestra su formato.

ITERACIÓN		INICIA		TERMINA			DURACIÓN			
1		14-ene-2013		18-ene-2013			5 días			
				L	M	M	J	V	S	
				14-ene	15-ene	16-ene	17-ene	18-ene	19-ene	
				Tareas	8	6	6	4	2	0
				Horas de trabajo pendientes	72	56	40	24	8	0
Requisito	Tarea	Miembro	Estado	Esfuerzo						
Requisito A	Tarea 1	Milton	Terminada	8						
Requisito A	Tarea 3	Gabriel	Terminada	8						
Requisito A	Tarea 5	Gabriel	Terminada	12	12	4				
Requisito B	Tarea 2	Milton	Terminada	12	12	4				
Requisito B	Tarea 6	Milton	Terminada	12	12	12	8			
Requisito C	Tarea 4	Gabriel	Terminada	8	8	8	4			
Requisito C	Tarea 7	Milton	En curso	8	8	8	8	4		
Requisito C	Tarea 8	Gabriel	No iniciada	4	4	4	4	4	0	

Figura III.52. Reserva de Iteración.

Fuente: Los Autores.

Dónde:

- **Requisito:** La Historia de Usuario a ser desarrollada en la Iteración.
- **Tarea:** Las tareas que fueron divididas de la Historia de Usuario y que su desarrollo tenga como máximo 16 horas.
- **Miembro:** El nombre del miembro del Equipo DAV responsable de desarrollar la tarea.
- **Estado:** Indica el estado que se encuentra la tarea durante la Iteración, siendo este: no iniciada, en curso o terminada.
- **Esfuerzo:** El esfuerzo que demanda la construcción de la tarea.

Características

- ✓ Contiene tanto los requerimientos como las tareas asociadas a cada uno de ellos que serán desarrolladas a lo largo de la Iteración.
- ✓ Permite llevar un registro del avance del Equipo DAV día a día.
- ✓ Siempre debe estar visible a todo el equipo.
- ✓ No se modifica el alcance durante la Iteración actual.
- ✓ Tiene gran importancia ser realista en la elaboración de la Reserva de Iteración para poder finalizar las tareas acordadas

2.1.5.10.4. Burn-Down

Es un gráfico que muestra cuánto le falta al Equipo DAV para completar con el compromiso de la Iteración. Sobre el eje horizontal se ubican los días hábiles que tiene la Iteración y sobre el eje vertical la cantidad de puntos que tienen que quemar durante esa Iteración (ver **Figura III.53.**).

La cantidad de puntos a quemar se calcula durante la Planificación de la Iteración, en donde el Equipo DAV se compromete ante el Dueño del Producto a realizar cierta cantidad de tareas. La suma de las estimaciones de estas tareas es la cantidad de puntos que el equipo deberá quemar durante la Iteración para cumplir con el compromiso.

El gráfico se actualiza diariamente, durante la Reunión Diaria de DAV, el equipo actualiza las estimaciones de las tareas, y así va contando cuántos puntos le quedan por terminar la Iteración.

El desvío que existe entre esta curva y la línea de quemado ideal nos indica qué tan bien está el equipo respecto a su planificación.

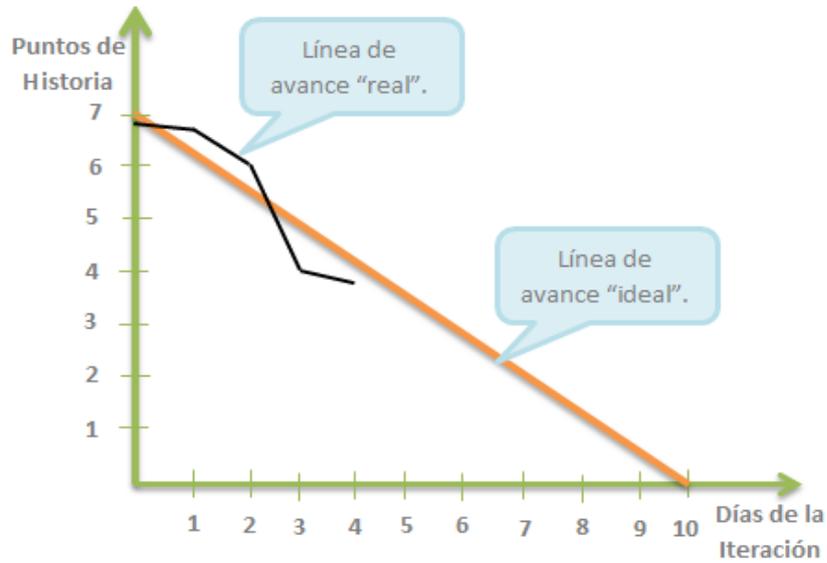


Figura III.53. Gráfico Burn-Down de DAV.
Fuente: Los Autores.

Cuando se estima los Puntos de Historia, lo que se intenta es dar un valor del tamaño de la Historia de Usuario implicando una combinación de:

- **Complejidad:** La dificultad que existe en el desarrollo de la Historia de Usuario.
- **Esfuerzo:** El tiempo que se demorará la Historia de Usuario en desarrollarse.
- **Riesgo:** Si el Equipo DAV tiene experiencia en las herramientas a utilizarse en el desarrollo del videojuego.

2.1.5.10.5. Burn-Up

Es un gráfico que muestra cuánto avanzó el Equipo DAV hasta el momento, cuántos puntos llegaron a quemar hasta la última Iteración, en relación al esfuerzo total necesario para terminar el proyecto, permitiendo ver que tanto cambio el alcance en el tiempo, el valor ganado y el tiempo faltante para las fechas de entregas.

Es ideal para reportar estados del proyecto y explicar al Dueño del Producto en que se está poniendo esfuerzo y que resultado se está obteniendo.

Sobre el eje horizontal se ubican las Iteraciones, sobre el eje vertical se ubican los puntos de Historias (ver **Figura III.54.**).

Al inicio se estiman las Historias de Usuario que el Dueño del Producto tiene, con lo cual se establece el "techo" del Burn-Up. Sabiendo el factor de foco del Equipo DAV y su cantidad de integrantes podemos saber su velocidad (cuántos puntos quemarán en cada Iteración). Con estos datos se puede trazar la "línea de quemado ideal", la cual indica el avance ideal para llegar a quemar todos los puntos del proyecto.

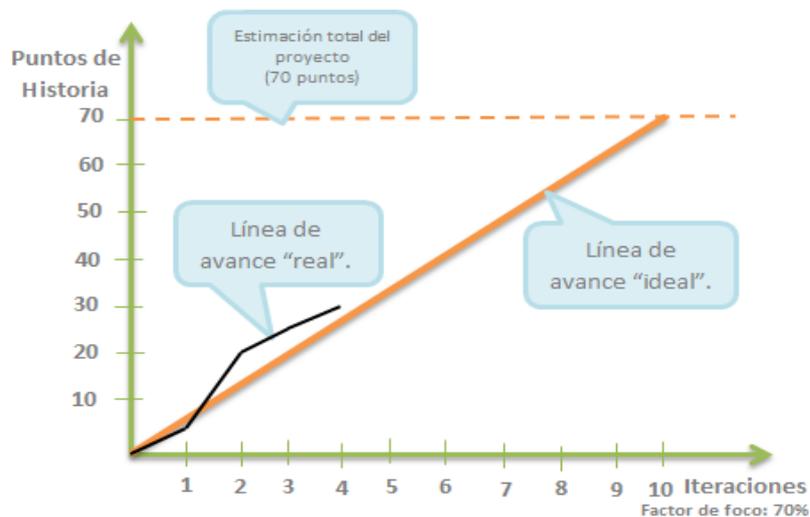


Figura III.54. Gráfico de Burn-Up.

Fuente: Los Autores.

El cálculo se realiza de la siguiente manera:

Velocidad = (Nº desarrolladores del equipo * días de duración de la Iteración) * Factor de Foco %.

Dónde:

Factor de Foco (FF%) es el % del día que los miembros del Equipo DAV son plenamente productivos y pueden estar dedicados al 100% a la producción del videojuego, sin verse afectados por otros impedimentos, reuniones o tareas auxiliares.

Suele fijarse de partida entre el 60% y el 70 %, dependiendo de la madurez del equipo, y se va recalculando iteración tras iteración [²⁴].

2.1.5.10.6. Cartas DAV

Las Cartas DAV sirven para estimar la duración de las Historias de Usuario. Consiste básicamente en que cada miembro del Equipo DAV disponga de una baraja de cartas. Estas cartas no son cartas corrientes, aunque podría utilizarse una baraja estándar. Una vez que el DAV Máster exponga las especificaciones de una Historia de Usuario, se procede a la votación. Cada miembro del Equipo DAV debe seleccionar las cartas que sumen el esfuerzo estimado para realizar la tarea (en horas, puntos de historia, o la medida que use el equipo). Cuando todos los miembros del equipo han estimado se da la vuelta a las cartas a la vez, y todos muestran su estimación.

El DAV Máster, inspecciona las valoraciones y se fija en las más extremas por arriba y por abajo, y pregunta a los miembros del equipo que expliquen sus razones para la valoración. Después de nuevo se vuelve a estimar con el mismo método, hasta que las valoraciones sean razonablemente parecidas, aunque también se puede determinar la estimación con otros criterios tales como la media de las valoraciones, o incluso criterios subjetivos del DAV Máster.

Las Cartas DAV se basa en la utilizada por Scrum (**ver Figura III.55.**), pero con algunos valores cambiados quedando de la siguiente manera: 0 para una tarea que ya está hecha, es fácilmente replicable o se tarda unos pocos minutos, 1/2, 1, 2, 4, 6, 8, 10, 12, 14, 16, “#” indica que el valor estimado no se encuentra en ninguna carta permitiendo asignar el valor verbalmente sin sobrepasar de 15, “?” que indica “no tengo ni idea” o “más o menos”, infinito para una estimación demasiado grande o extensa y una carta especial que representa un café o una jarra de cerveza que quiere decir “necesito un descanso”. De esta manera se limita la estimación a 16 horas que debería durar una tarea como máximo.

²⁴ <http://www.gestiondeproyectosit.es/blogit/2012/12/metricas-agiles-ii/>



Figura III.55. Cartas DAV.

Fuente: Los Autores.

Votación Secreta

No debe conocerse la puntuación de cada uno hasta que todos los miembros del Equipo DAV hayan realizado su votación para no influenciar el voto de los demás, debido a que si alguien vota, por ejemplo, 13, y otros dos compañeros anteriores han votado 5, esto le puede hacer caer en la tentación de votar también 5 u 8 a lo sumo, para no parecer más tonto o algo parecido.

De esta forma, al ser secreta cada miembro votará lo que realmente piensa. Cuando el DAV Máster realice el recuento, podrá ver fácil y rápidamente si alguno o varios miembros del Equipo DAV no han entendido bien las especificaciones de la Historia de Usuario, ya sea por exceso o carencia de tiempo y por lo tanto de dificultad, procediendo de esta forma a la re explicación de la Historia de Usuario y sus especificaciones para repetir las votaciones hasta que se llegue a un consenso.

Características

- ✓ Permite agilizar las reuniones en las que se estiman tareas, haciéndolas más dinámicas y divertidas.
- ✓ Implica a todo el equipo en el proceso de estimar, aportando mucho más valor.

- ✓ Estimación basada en consenso.
- ✓ Distribución de la responsabilidad en la estimación.
- ✓ Todos acaban entendiendo los pormenores de cada tarea.
- ✓ Se sacan a la luz códigos de ejemplo/ideas ya existentes que ayudaran a desarrollar más rápido.

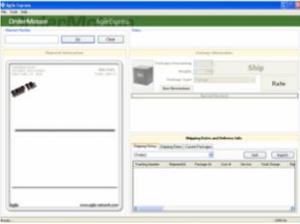
2.1.6. Herramientas

Para la gestión y seguimiento de un proyecto DAV se pueden utilizar las herramientas de Scrum, ayudando a manejar problemas que surjan en el transcurso del desarrollo o liberación del videojuego, permitiendo gestionar la Reserva del Producto, la Reserva de Iteración y las Historias de Usuario, además permiten un seguimiento de la velocidad y avance de las Iteraciones. Para la refactorización de código se puede utilizar las herramientas que usa XGD.

En la siguiente tabla se detalla algunas herramientas que se pueden utilizar.

Tabla III.XLV. Herramientas que pueden ser utilizadas por DAV.

HERRAMIENTAS QUE PUEDEN SER UTILIZADAS POR DAV	DESCRIPCIÓN
<p>SPRINTTOMETER</p> 	<p>Herramienta simple y fácil de usar pensada para llevar a cabo la gestión, métrica y seguimiento de proyectos ágiles de una manera sencilla, con el fin de simplificar el intercambio de datos con programas externos, todas las hojas de cálculo se pueden exportar a Microsoft Excel, también se puede utilizar el formato XML para los archivos locales.</p> <p>Algunas de las principales características de este software son:</p> <ul style="list-style-type: none">• Admite el tamaño variable del equipo• Licencia Libre.• Asignación de recursos a las tareas y los historiales de usuario.• Separa el seguimiento de las Tareas de desarrollo y pruebas• Cálculo del ámbito diario para que los desarrolladores y evaluadores estén al tanto.• Seguimiento del proyecto de presupuesto protegido

	<p>con contraseña</p>
<p>EASYBACKLOG</p> 	<p>Herramienta muy intuitiva para la gestión de Tareas, facilidad de uso de su interfaz, los paneles de métricas y estadísticas de los que dispone, así como la facilidad de asignar Tareas a cada una de las Iteraciones, reduce el tiempo dedicado a la gestión de reservas, controla el exceso del proyecto, proporciona informes en tiempo real y simplifica la gestión de la Iteración.</p> <p>Sus principales características son:</p> <ul style="list-style-type: none"> • Simplicidad en la interfaz. • Licencia libre. • Velocidad de crear un nuevo atraso (el modelo de cómo funciona Excel). • Las instantáneas le permiten trabajar de una manera ágil, manteniendo un seguimiento de cómo está evolucionando la reserva para que pueda informar sobre los impactos de costos o el tiempo a sus clientes.
<p>AGILE EXPRESS</p> 	<p>Es una herramienta libre que proporciona módulos para gestionar y registrar Historias de Usuario y Tareas, proyectos e iteraciones. A su vez contiene un tablero virtual que muestra el progreso de las Historias de Usuario y Tareas. Ofrece también la posibilidad de generar gráficos Burn-Down y comparar la velocidad entre iteraciones.</p>
<p>FIRESCRUM</p> 	<p>Es una herramienta libre desarrollada para apoyar la gestión de proyectos bajo la metodología ágil Scrum. Es una aplicación Web intuitiva y fácil de usar. Contiene la técnica de estimación de Planning Poker con audio y video, además un módulo de gestión de incidencias.</p>
<p>ICESCRUM</p> 	<p>Herramienta libre de Scrum. Ofrece las opciones de operación, consulta y estimación de Historias de Usuario. Permite añadir Historias de Usuario a la Reserva del Producto, dividir el tiempo en Iteraciones y mover las Historias de la Reserva del Producto a cada una de las Iteraciones. Posee la técnica de Planning Poker para la estimación y paneles virtuales.</p>
<p>KUNAGI</p>	<p>Herramienta Web gratuita que proporciona un sistema integral de gestión de proyectos. Ofrece herramientas colaborativas y otras facilidades como un cuadro de mando del proyecto, panel interactivo para la Iteración actual o la técnica de estimación de</p>

	Poker.
<p>ECLIPSE</p> 	<p>Es utilizado por XGD debido a que una de las herramientas más potentes con las que cuenta es la de refactorización.</p> <p>Sus principales características son:</p> <ul style="list-style-type: none">• Compilación en tiempo real.• Dispone de un editor de texto con resaltado de sintaxis.• Posee pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización.• Permite la integración con Hibernate.

Elaborado por: Los Autores.

CAPÍTULO IV

4. DESARROLLO DE UN VIDEOJUEGO

En el presente capítulo se explica el diseño y desarrollo de un videojuego utilizando la metodología adaptada DAV.

4.1. FASE PRE-JUEGO

4.1.1. Descripción del videojuego

El videojuego a desarrollar se lo ha llamado “El Tren del Conocimiento”, no se ha basado en ningún videojuego, su idea nace por iniciativa propia de los autores de esta tesis.

El videojuego será en 2D y tendrá un personaje principal que es un tren, el mismo será controlado por el jugador el cual debe llevarlo por las rieles o caminos, sin dejarlo salir he ir cogiendo los objetos que van apareciendo, si el tren sale de la vía o colisiona con sus vagones perderá el juego; al coger 1 objeto la puntuación se incrementará 10 puntos, al coger 3 objetos se agregará un vagón al tren y al coger 6 objetos aparecerá

una pregunta referente a la Provincia de Chimborazo, si esta es contestada correctamente se incrementará 10 puntos, caso contrario se restará 10 puntos a la puntuación. Cuando se tenga agregado 10 vagones al tren y hayan aparecido 5 preguntas, el nivel será completado habilitando el siguiente nivel. El videojuego constará de 8 niveles, los mismos estarán deshabilitados, a excepción del primer nivel, cuando se completen todos los niveles el juego terminará exitosamente, dando la posibilidad en cada nivel de registrar los datos del jugador para agregar la puntuación obtenida en cada juego.

4.1.2. Aspectos Fundamentales

Genero

“El tren del Conocimiento” será un videojuego de Género Mixto debido a que poseerá características de los Géneros Puzzle al probar las habilidades y destrezas del jugador al conducir el tren sin dejarlo salir de los rieles o caminos cogiendo los objetos que irán apareciendo y Educativo porque transmitirá al jugador conocimiento sobre la Provincia de Chimborazo mediante la formulación de preguntas.

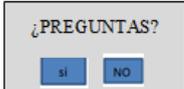
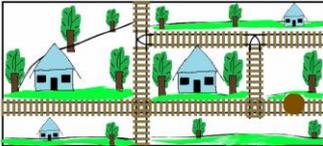
Historia

Con este videojuego se pretende lograr que el jugador mientras se divierte vaya adquiriendo conocimiento con preguntas planteadas sobre la Provincia de Chimborazo, para lograr esto el videojuego estará basado en un tren como personaje principal, el cual será controlado por el jugador, llevándolo por los rieles o caminos sin dejarlo salir o colisionar con sus vagones cogiendo los objetos que van apareciendo, de esta manera las preguntas aparecerán cuando se coja un cierto número de objetos.

Bocetos

En la siguiente tabla se muestra los bocetos o diseños preliminares de los elementos que conformarán el videojuego.

Tabla IV.XLVI. Diseños preliminares del Videojuego “El tren del Conocimiento”.

ELEMENTOS	DISEÑO	DESCRIPCIÓN
Tren		Será el personaje principal el cual el jugador controlará.
Vagones		Irán incorporándose en el tren cada vez que se cojan 3 objetos.
Controles		Permitirán al jugador manipular al tren.
Objetos		Existirán 6 objetos diferentes que irán apareciendo en el camino del tren, servirán para superar cada nivel, cada cierto número de objetos cogidos se incorporará un vagón al tren y aparecerá una pregunta.
Preguntas		Serán invocadas luego de coger 6 objetos y mediante las cuales se pretende que el jugador adquiera conocimiento sobre la Provincia de Chimborazo.
Preguntas Correctas		Será invocada luego de responder correctamente una pregunta.
Preguntas Incorrectas		Será invocada luego de responder incorrectamente una pregunta, en donde se mostrará al jugador la respuesta correcta.
Mundo o Niveles		Será el escenario o el ambiente en donde transcurrirá el juego.

Elaborado por: Los Autores.

Aspectos

El videojuego será desarrollado utilizando la técnica 2D y tendrá una vista Cenital es decir la posición de la cámara será vista aérea, en la siguiente tabla se muestra los colores que tendrán los elementos del videojuego.

Tabla IV.XLVII. Aspectos del Videojuego “El tren del Conocimiento”.

ELEMENTOS	COLORES
Tren	La mayor parte del tren poseerá un color rojo, en la parte superior un color amarillo, los lados tendrán un color plomo con franjas negras y la trompa del tren será negra.
Vagones	Serán de color rojo totalmente.
Controles	Su fondo será de color azul con flechas blancas dirigidas en un solo sentido.
Objetos	Algunos objetos serán dorados y otros de color café.
Preguntas	Tendrán un fondo degradado de color celeste y sus letras serán de color amarillo.
Preguntas Correctas	En el centro de la imagen tendrá un visto de color verde y un botón “continuar” de color azul con letras blancas.
Preguntas Incorrectas	En el centro de la imagen tendrá una x de color rojo, un botón “continuar” de color azul con letras blancas y las respuestas con un título de color celeste con su contenido con letras amarillas
Mundo o Niveles	Algunos mundos o niveles tendrán el aspecto de un paisaje verde con flora, de desierto y otros con aspecto de paisaje de hielo, con rieles del tren de color plomo con soportes de color café.

Elaborado por: Los Autores.

Interfaz de Usuario

El control del personaje principal (Tren) se realizará a través de la pantalla táctil. En el escenario se encontrarán ubicados dos botones uno en el extremo derecha y otro en el extremo izquierda, dependiendo para donde se quiere que el personaje se desplace se presionará dicho botón y este lo rotará 90 grados ya sea a la derecha o izquierda de donde se encuentre.

Objetivo

El objetivo o meta del videojuego es coger con el tren los objetos que van apareciendo en los rieles o caminos y responder correctamente las preguntas planteadas a lo largo de los niveles para conseguir el máximo número de puntos posibles. Los puntos son otorgados cuando cogemos los objetos y contestamos correctamente las preguntas.

Reglas

- El personaje (Tren) podrá desplazarse solo por los rieles o caminos marcados, si este se sale perderá el juego.
- El personaje (Tren) cuando tenga menos de 2 vagones podrá girar más de 90 grados en cualquier dirección para desplazarse.
- El personaje (Tren) cuando tenga más de 1 vagón no podrá girar más de 90 grados en cualquier dirección, si lo hiciera colisionará con los vagones y perderá el juego.
- El personaje deberá coger los objetos que van apareciendo en los rieles o caminos para seguir avanzando y terminar exitosamente el juego.
- El jugador deberá obligadamente responder a las preguntas que se plantearán en los distintos niveles.

Características

En la siguiente tabla se especifica las principales características de cada elemento del videojuego.

Tabla IV.XLVIII. Diseños preliminares del Videojuego “El tren del Conocimiento”.

ELEMENTOS	DESCRIPCIÓN
Tren	Se mostrará la parte superior para dar la forma de una vista aérea, con un efecto de ir votando humo.
Vagones	Su forma será cuadrada, dando el espeto de una vista aérea.
Controles	Los botones de los controles tendrán la forma circular con tres flechas en su interior dirigidas en dirección del lado que funcionan respectivamente.
Objetos	Existirán 6 objetos con la forma de hongos, cofre y monedas, diferenciándose cada uno.
Preguntas	Cada pregunta contendrá información importante de la Provincia de Chimborazo, ya sea correcta o incorrecta, dando la posibilidad de que el jugador al contestar afirme o niegue mediante los botones (SI, NO).
Preguntas Incorrectas	Tendrá la respuesta correcta de una pregunta planteada falsa, será presentada al contestar un pregunta incorrectamente, además tendrá un botón en el centro de la pantalla para continuar con el juego.
Mundo o Niveles	Cada nivel o escenario tendrá un diferente camino o trazado con aspectos tales como: paisajes verdes con flora, desiertos y paisajes de hielo.

Elaborado por: Los Autores.

Diseño de niveles

El videojuego estará compuesto de 8 niveles, cada uno tendrá un camino o trazado diferente convirtiéndose en un reto y aumentando la dificultad a medida que se supere un nivel, además en cada uno se plantearán diferentes preguntas como parte del reto.

Requerimientos Técnicos

El videojuego será desarrollado en la versión de Android 2.3.3 y podrá reproducirse en el máximo número de versiones de Android disponibles, como mínimo se necesitará la versión 2.0 y móviles que tengan cualquier tipo de pantalla táctil con un espacio de almacenamiento disponible de 25 MB. Consiguiendo, por tanto, la compatibilidad con la mayoría de dispositivos.

4.1.3. Arquitectura del Videojuego

En la siguiente figura se muestra la arquitectura que tendrá el videojuego.

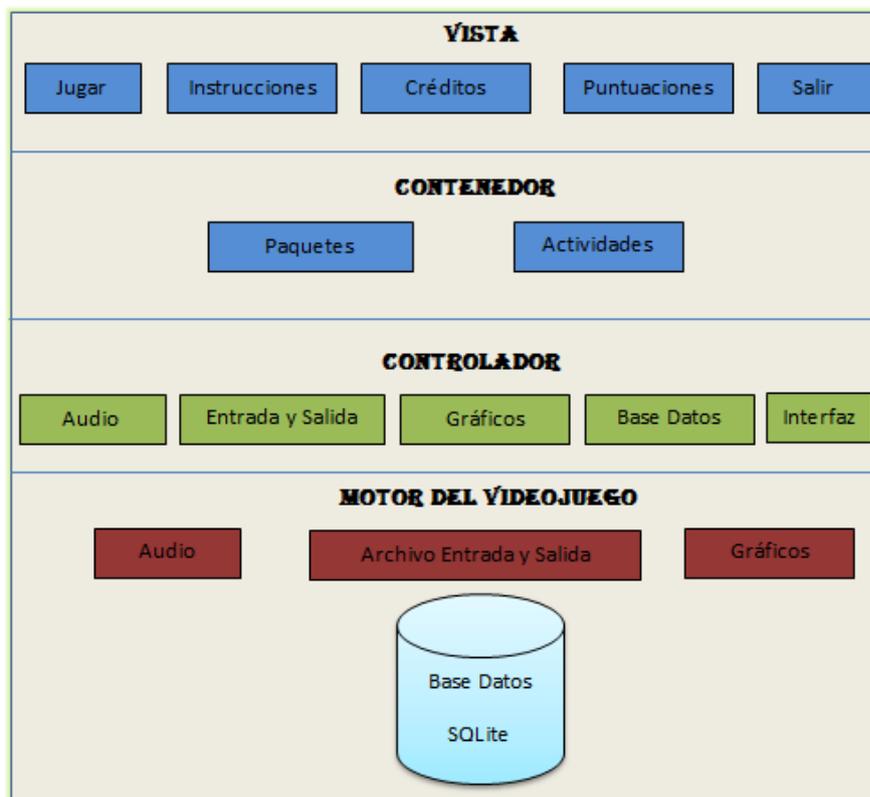


Figura IV.56. Arquitectura del videojuego “El Tren del Conocimiento”.
Fuente: Los Autores.

4.1.4. Diseño Conceptual

En la siguiente figura se muestra el diseño conceptual del videojuego, una vez recopilada la información necesaria para su desarrollo.

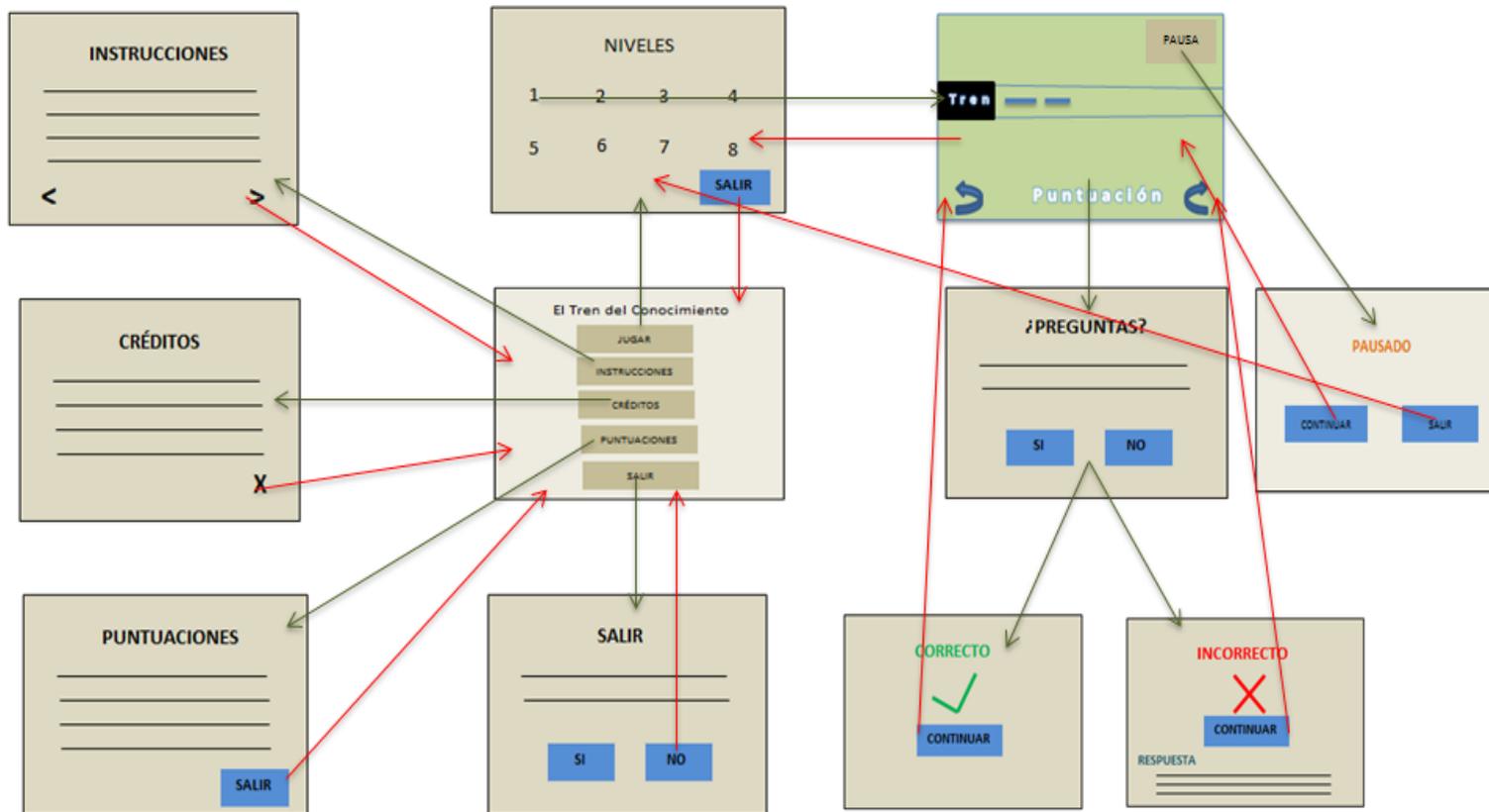


Figura IV.57. Diseño Conceptual del videojuego “El Tren del Conocimiento”.

Fuente: Los Autores.

4.1.5. Herramientas a Utilizarse

Para el diseño y desarrollo del videojuego se utilizarán las siguientes herramientas:

Diseño

- **Pixlr:** Es un editor online gratuito de imágenes el cual nos servirá para crear y editar los elementos del videojuego tales como: el personaje principal (Tren), los vagones, los objetos (monedas, cofre y hongos), los mundos o niveles (camino o rieles, casas, árboles, etc.).
- **The Logo Creator:** Este programa se utilizará para crear el logo principal del videojuego de un forma simple y eficaz.
- **Cool Text:** Es un generador de texto y logos online gratuito el cual será utilizado para crear los diferentes textos y botones utilizados en cada pantalla del videojuego.
- **SonidosGratis:** Es una página web en donde podremos descargar gratuitamente todo tipo de sonido para el videojuego.
- **aTube Catcher:** Este programa se utilizará para recortar los diferentes sonidos que serán parte del videojuego.

Desarrollo

- Para el desarrollo se utilizará como IDE el entorno de desarrollo Eclipse con el plugin ADT (Android Developer Tools) de android el cual incluye componentes esenciales como Android SDK (Kit de desarrollo de software) y la herramienta AVD (Android Virtual Device) Manager, la cual que permitirá emular en una computadora los dispositivos móviles a los que apunta nuestro videojuego y de esta manera realizar las pruebas de funcionamiento.

4.1.6. Roles

Para el desarrollo del videojuego se asignaron los siguientes roles mostrados en la **Tabla IV.XLIX.**, distribuidos únicamente entre los dos autores de esta tesis.

Tabla III.XLIX. Roles asignados para el desarrollo del videojuego “El Tren del Conocimiento”.

Persona	Contacto	Rol
Gabriel Armendáriz	garmy121t@yahoo.es /0987382825	DAV Master
Milton Saltos	milton3552@hotmail.com /0984752026	Dueño del Producto
Gabriel Armendáriz	garmy121t@yahoo.es /0987382825	Equipo DAV (Sonidista)
Milton Saltos	milton3552@hotmail.com /0984752026	Equipo DAV (Diseñador)
Gabriel Armendáriz	garmy121t@yahoo.es /0987382825	Equipo DAV (Diseñador)
Gabriel Armendáriz	garmy121t@yahoo.es /0987382825	Equipo DAV (Programador)
Milton Saltos	milton3552@hotmail.com /0984752026	Equipo DAV (Programador)
Milton Saltos	milton3552@hotmail.com /0984752026	Equipo DAV (Tester)

Elaborado por: Los Autores.

4.1.7. Artefactos

Historias de Usuario

Las siguientes Historias de Usuario mostradas en la **Tabla IV.L.**, presentan las funcionalidades que tendrá el videojuego “El Tren del Conocimiento”.

Tabla III.L. Historias de Usuario del Videojuego DAV.

ID	Como	Quiero / Quiero que...	De modo que / Para
HU01	Jugador	Tener un menú principal	Pueda ingresar a jugar, visualizar las instrucciones, créditos, puntuaciones y salir del videojuego.
HU02	Jugador	Se muestre un splash al iniciar la aplicación (videojuego).	Sea divertido.
HU03	Jugador	Los objetos aparezcan únicamente en la vía del tren.	Se puedan ir cogiendo.
HU04	Jugador	Visualizar las instrucciones del juego	Saber cómo jugar.
HU05	Jugador	Visualizar los créditos del juego	Saber quién lo desarrolló.
HU06	Jugador	Visualizar las 10 primeras altas puntuaciones.	Saber cuáles jugadores la sacaron.

HU07	Jugador	Los controles del tren se encuentren en la parte inferior de la pantalla distribuidos en los extremos respectivamente.	Pueda facilitar la jugabilidad.
HU08	Jugador	El tren pueda moverse solo por los rieles o caminos.	Si se saliese de la vía termine el juego.
HU09	Jugador	El tren pueda girar más de 90 grados con menos de 2 vagones	Pueda darse la vuelta para cambiar el rumbo.
HU10	Jugador	El tren no pueda girar más de 90 grados	Si lo hiciera chocara con sus vagones y terminara el juego.
HU11	Jugador	El tren al coger los objetos se escuche un sonido de “incremento”.	Poder dar realismo al juego.
HU12	Jugador	Existan 8 niveles los mismos que se encuentren deshabilitados exceptuando el primero.	Al ir completando un nivel se vaya deshabilitando el siguiente.
HU13	Jugador	Ingresar mi nombre y apellido y una foto si lo deseara.	Poder visualizar junto a la puntuación alcanzada.
HU14	Jugador	Visualizar el puntaje al coger los objetos y al responder correctamente las preguntas.	Pueda conocer la puntuación que voy alcanzando.
HU15	Jugador	Pausar el juego al pulsar un botón.	Poder hacer algo en ese momento o salir del juego.
HU16	Jugador	Cancelar el ingreso de datos.	No visualizar la puntuación.
HU17	Jugador	El tren aumente los vagones a medida que vaya cogiendo 3 objetos.	Aumentar la diversión del juego.
HU18	Jugador	Las preguntas vayan apareciendo a medida que vaya cogiendo 6 objetos.	Aumentar la diversión del juego.
HU19	Jugador	Antes empezar a jugar se visualice el nivel del juego en una pantalla de preparado	Sea más óptimo.
HU20	Jugador	Pulsar en cualquier parte de la pantalla preparado.	Empezar a jugar.
HU21	Jugador	Visualizar una pantalla de “Fin del Juego” al perder el juego conjuntamente con dos botones de “Nuevo Juego” y “Salir”	Saber que perdí y poder intentarlo de nuevo o salir del juego.
HU22	Jugador	Poder responder las preguntas que se van mostrando mediante dos botones de afirmación (SI) o negación (NO).	Poder saber el grado de conocimiento que tengo sobre la Provincia de Chimborazo y adquirir nuevo conocimiento.

HU23	Jugador	En cada nivel se muestren como máximo 5 preguntas.	El juego no sea aburrido.
HU24	Jugador	Al escoger en el menú principal la opción salir se muestre una pantalla de confirmación de cerrar la aplicación.	Si no estoy seguro de salir pueda regresar al menú principal o simplemente cerrar la aplicación.
HU25	Jugador	Existan 2 botones para poder acceder a las redes sociales del Facebook o Twitter.	Conocer opiniones del juego e información adicional del mismo.
HU26	Jugador	Exista un botón de sonido en la pantalla de menú principal.	Pueda habilitar y deshabilitar el sonido.
HU27	Sonidista	Exista sonido cuando el tren este en movimiento	Dar un aspecto real al juego.
HU28	Diseñador	El tren tenga un aspecto real	Se muestre más divertido.
HU29	Diseñador	Los vagones tengan un aspecto de un cuadrado de color rojo.	Se muestre realista.
HU30	Diseñador	Los objetos tengan un aspecto de cofres, hongos y monedas de colores dorados y cafés.	Se muestren divertidos.
HU31	Diseñador	Existan animaciones en la pantalla del menú principal, créditos, instrucciones, niveles y salir	De un aspecto divertido.
HU32	Diseñador	Exista en la pantalla de créditos, instrucciones, puntuaciones, niveles y salir un botón de regreso.	Poder volver al menú principal.
HU33	Sonidista	Exista audio de fondo en las diferentes pantallas del videojuego.	Sea divertido.
HU34	Diseñador	La pantalla de puntuaciones muestre el nombre, apellido, puntuación, número de preguntas correctas e incorrectas contestadas de los 10 jugadores con más altas puntuaciones.	Se pueda apreciar mejor los datos de los jugadores.
HU35	Diseñador	Exista en la pantalla de Instrucciones botones de navegabilidad entre sus pantallas.	Poder navegar fácilmente.
HU36	Sonidista	Se escuche un sonido al iniciarse el Splash.	Se vea divertido.

HU37	Jugador	El contenido de las preguntas sean sobre la provincia de Chimborazo	Pueda complementar mi conocimiento.
HU38	Diseñador	Al responderse una pregunta correctamente se muestre un mensaje de “CORRECTO” con un visto de color verde, con un botón de “continuar” para seguir jugando, conjuntamente con un sonido de correcto.	Pueda saber que se respondió correctamente.
HU39	Diseñador	Al responderse una pregunta incorrectamente se muestre un mensaje de “INCORRECTO” con una x de color rojo, un botón de “continuar” para seguir jugando, conjuntamente con un sonido de incorrecto.	Pueda saber que se respondió incorrectamente.
HU40	Jugador	Al responder una pregunta incorrectamente se muestre la respuesta correcta.	Pueda saber que es lo correcto.
HU41	Jugador	La puntuación del juego se vaya acumulando nivel tras nivel	Conocer el total alcanzado al finalizar el juego.
HU42	Jugador	La puntuación disminuya cuando se responda incorrectamente en 10 puntos	No pueda alcanzar la máxima puntuación.
HU43	Jugador	Al completar todos los niveles y al responder correctamente todas las preguntas, se muestre una pantalla de congratulación.	Pueda sentirme elogiado.
HU44	Jugador	Al completar un nivel correctamente, se muestre una pantalla de nivel completado, con un botón de “CONTINUAR” y otro de “SALIR”.	Pueda saber que termine correctamente el nivel.

Elaborado por: Los Autores.

Pruebas de Aceptación

En este apartado se muestran las Pruebas de Aceptación, las mismas que validaran las diferentes funcionalidades del videojuego una vez desarrolladas, de esta manera se garantizará que cumplan con lo requerido y pasarán al estado de completado. (**Ver Anexo II**).

Prueba de Aceptación	
<i>Identificador: PA01</i>	<i>Historia de Usuario (Nro. Y Nombre): HU01-Como jugador quiero tener un menú principal.</i>
<i>Nombre: Realizar el menú principal.</i>	
<i>Descripción: Realizar un menú principal en donde se pueda escoger Jugar, Instrucciones, Créditos, Puntuaciones y Salir</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil en completo funcionamiento.</i>	
<i>Entrada / Pasos de ejecución:</i> 1. Presentar el menú principal de color rojo degradado con sus respectivas opciones.	
<i>Resultado Esperado: Mostrar las diferentes opciones (pantallas) al dar clic sobre una de ellas.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Figura IV.58. Prueba de Aceptación “PA01”.

Fuente: Los Autores.

Reserva del Producto

En la siguiente tabla se observan las Historias de Usuario con sus respectivas Pruebas de Aceptación ya estimadas mediante las cartas DAV y priorizadas por el Dueño del Producto.

Tabla IV.LI. Reserva del Producto del videojuego “El Tren del Conocimiento”.

Prioridad	Nro. Historia de Usuario	Ítem	Estimación	Estado	Prueba de aceptación
3	HU01	Como Jugador, quiero tener un menú principal de modo que pueda ingresar a jugar, visualizar las instrucciones, créditos, puntuaciones y salir de videojuego.		Pendiente	PA01
2	HU02	Como Jugador, quiero que se muestre un splash al iniciar la aplicación (videojuego), de modo que sea divertido.		Pendiente	PA02
1	HU03	Como Jugador, quiero que los objetos aparezcan únicamente en la vía del tren, de modo que se puedan ir cogiendo.		Pendiente	PA03
3	HU04	Como Jugador quiero visualizar las instrucciones del juego, para saber cómo jugar.		Pendiente	PA04
3	HU05	Como Jugador, quiero visualizar los créditos del juego, para saber quién lo desarrollo.		Pendiente	PA05
4	HU06	Como Jugador, quiero visualizar las 10 primeras altas puntuaciones, de modo que pueda saber cuáles jugadores la obtuvieron.		Pendiente	PA06
1	HU07	Como Jugado, quiero que los controles del tren se encuentren en la parte inferior de la pantalla distribuidos en los extremos respectivamente, para que pueda facilitar la jugabilidad.		Pendiente	PA07

1	HU08	Como Jugador, quiero que el tren pueda girar más de 90 grados con menos de 2 vagones, de modo que pueda darse la vuelta para cambiar el rumbo		Pendiente	PA08
1	HU09	Como Jugador, quiero que el tren no pueda girar más de 90 grados, de modo que si lo hiciera chocara con sus vagones y terminara el juego.		Pendiente	PA09
1	HU10	Como Jugador, quiero que el tren pueda moverse solo por los rieles o caminos, de modo que si se saliese de la vía termine el juego.		Pendiente	PA10
3	HU11	Como Jugador, quiero que el tren al coger los objetos se escuche un sonido de “incremento”, para poder dar realismo al juego.		Pendiente	PA11
5	HU12	Como Jugador, quiero que existan 8 niveles los mismos que se encuentren deshabilitados exceptuando el primero, de modo que al ir completando un nivel se vaya deshabilitando el siguiente.		Pendiente	PA12
5	HU13	Como Jugador, quiero Ingresar mi nombre y apellido y una foto si lo deseara, para poder visualizar junto a la puntuación alcanzada.		Pendiente	PA13
2	HU14	Como Jugador, quiero visualizar el puntaje al coger los objetos y al responder correctamente las preguntas, de modo que pueda conocer la puntuación que voy alcanzando.		Pendiente	PA14
2	HU15	Como Jugador, quiero pausar el juego al pulsar un botón, de modo que pueda hacer algo en ese momento o salir del juego.		Pendiente	PA15
5	HU16	Como Jugador, quiero cancelar el ingreso de datos, para no visualizar la puntuación.		Pendiente	PA16

5	HU17	Como Jugador, quiero que el tren aumente los vagones a medida que vaya cogiendo 3 objetos, para aumentar la diversión del juego.		Pendiente	PA17
4	HU18	Como Jugador, quiero que las preguntas vayan apareciendo a medida que vaya cogiendo 6 objetos, para aumentar la diversión del juego.		Pendiente	PA18
2	HU19	Como Jugador, quiero que antes de empezar a jugar se visualice el nivel del juego en una pantalla de preparado, para que sea más óptimo.		Pendiente	PA19
2	HU20	Como Jugador, quiero pulsar en cualquier parte de la pantalla preparado., para empezar a jugar		Pendiente	PA20
4	HU21	Como Jugador, quiero visualizar una pantalla de “Fin del Juego” al perder el juego conjuntamente con dos botones de “Nuevo Juego” y “Salir”, para saber que perdí y poder intentarlo de nuevo o salir del juego.		Pendiente	PA21
4	HU22	Como Jugador, quiero poder responder las preguntas que se van mostrando mediante dos botones de afirmación (SI) o negación (NO), para poder saber el grado de conocimiento que tengo sobre la Provincia de Chimborazo y adquirir nuevo conocimiento.		Pendiente	PA22
5	HU23	Como Jugador, quiero que en cada nivel se muestren como máximo 5 preguntas, para que el juego no sea aburrido.		Pendiente	PA23
3	HU24	Como Jugador, quiero que al escoger en el menú principal la opción salir se muestre una pantalla de confirmación de cerrar la aplicación., de modo que si no estoy seguro de salir pueda regresar al menú principal o simplemente cerrar la aplicación.		Pendiente	PA24
4	HU25	Como Jugador, quiero que existan 2 botones para poder acceder a las redes sociales del Facebook o Twitter, para conocer opiniones del juego		Pendiente	PA25

		e información adicional del mismo.			
3	HU26	Como Jugador, quiero que exista un botón de sonido en la pantalla de menú principal, para poder habilitar y deshabilitar el sonido.		Pendiente	PA26
1	HU27	Como Sonidista, quiero que exista sonido cuando el tren este en movimiento, para dar un aspecto real al juego,		Pendiente	PA27
2	HU28	Como Diseñador, quiero que el tren tenga un aspecto real, de modo que se muestre más divertido.		Pendiente	PA28
3	HU29	Como Diseñador, quiero que los vagones tengan un aspecto de un cuadrado de color rojo, de modo que se muestre realista.		Pendiente	PA29
2	HU30	Como Diseñador, quiero que los objetos tengan un aspecto de cofres, hongos y monedas de colores dorados y cafés, de modo que se muestren divertidos.		Pendiente	PA30
5	HU31	Como Diseñador, quiero que existan animaciones en la pantalla del menú principal, créditos, instrucciones, niveles y salir, de modo que den un aspecto divertido.		Pendiente	PA31
4	HU32	Como Diseñador, quiero que exista en la pantalla de créditos, instrucciones, puntuaciones, niveles y salir un botón de regreso, de modo que pueda volver al menú principal.		Pendiente	PA32
3	HU33	Como Sonidista, quiero que exista audio de fondo en las diferentes pantallas del videojuego, de modo que sea divertido.		Pendiente	PA33
5	HU34	Como Diseñador, quiero la pantalla de puntuaciones muestre el nombre, apellido, puntuación, número de preguntas correctas e incorrectas		Pendiente	PA34

		contestadas de los 10 jugadores con más altas puntuaciones, de modo que se pueda apreciar mejor los datos de los jugadores			
4	HU35	Como Diseñador, quiero que exista en la pantalla de Instrucciones botones de navegabilidad entre sus pantallas, para poder navegar fácilmente.		Pendiente	PA35
6	HU36	Como Sonidista, quiero que se escuche un sonido al iniciarse el Splash, de modo que sea divertido.		Pendiente	PA36
5	HU37	Como Jugador, quiero que el contenido de las preguntas sea sobre la provincia de Chimborazo, de modo que pueda complementar mi conocimiento.		Pendiente	PA37
6	HU38	Como Diseñador, quiero que al responderse una pregunta correctamente se muestre un mensaje de “CORRECTO” con un visto de color verde, con un botón de “continuar” para seguir jugando, conjuntamente con un sonido de correcto., créditos, instrucciones, niveles, y salir del juego, de modo que pueda saber que se respondió correctamente.		Pendiente	PA38
6	HU39	Como Diseñador, quiero que al responderse una pregunta incorrectamente se muestre un mensaje de “INCORRECTO” con una x de color rojo, un botón de “continuar” para seguir jugando, conjuntamente con un sonido de incorrecto, de modo que pueda saber que se respondió incorrectamente.		Pendiente	PA39
6	HU40	Como Jugador, quiero que al responder una pregunta incorrectamente se muestre la respuesta correcta, de modo pueda saber que es lo correcto.		Pendiente	PA40
6	HU41	Como Jugador, quiero la puntuación del juego se vaya acumulando nivel tras nivel, para conocer el total alcanzado al finalizar el juego.		Pendiente	PA41

6	HU42	Como Jugador, quiero que la puntuación disminuya cuando se responda incorrectamente en 10 puntos, de modo que no pueda alcanzar la máxima puntuación,		Pendiente	PA42
6	HU43	Como Jugador, quiero que al completar todos los niveles y al responder correctamente todas las preguntas, se muestre una pantalla de congratulación, de modo que pueda sentirme elogiado.		Pendiente	PA43
6	HU44	Como Jugador, quiero que al completar un nivel correctamente se muestre una pantalla de nivel completado, con un botón de “CONTINUAR” y otro de “SALIR”, de modo que pueda saber que complete el nivel correctamente.	2	Pendiente	PA44

Elaborado por: Los Autores.

Determinación de las fechas de entrega

Toda la Reserva del Producto o funcionalidades del videojuego se han dividido en 6 iteraciones o entregas, cada una con una duración de una semana, en la **Tabla IV.LII.**, se muestra las fechas de inicio así como también las fechas de revisión de cada iteración. Para dividir la Reserva del Producto se basó en las estimaciones realizadas, sin recargar las tareas al Equipo DAV para poder cumplir con éxito los objetivos de cada entrega.

Tabla IV.LII. Fechas de entrega de cada Iteración “El Tren del Conocimiento”.

Número Iteración	Fecha Inicio	Fecha Fin	Fecha Entrega
Iteración1	25 de Marzo de 2013	29 de Marzo de 2013	30 de Marzo de 2013
Iteración2	01 de Abril de 2013	05 de Abril de 2013	06 de Abril de 2013
Iteración3	08 de Abril de 2013	12 de Abril de 2013	13 de Abril de 2013
Iteración4	15 de Abril de 2013	19 de Abril de 2013	20 de Abril de 2013
Iteración5	22 de Abril de 2013	26 de Abril de 2013	27 de Abril de 2013
Iteración6	29 de Abril de 2013	03 de Mayo de 2013	04 de Mayo de 2013

Elaborado por: Los Autores.

4.2. FASE JUEGO

Una vez definido el alcance que tendrá el videojuego “El Tren del Conocimiento” y teniendo una idea clara de todo lo que se debe hacer se empieza con el desarrollo.

4.2.1. Iteraciones

La duración de cada iteración será de 5 días es decir de lunes a viernes, los días lunes se realizaran las Reuniones de Planificación de Iteración y las revisiones de las mismas se efectuarán los días sábados, de esta manera los otros días serán usados para desarrollar las funcionalidades de cada entrega. Para cumplir con el objetivo de cada iteración la duración de las reuniones diarias serán de 15 minutos.

Iteración 1

El objetivo de esta iteración es obtener la primera versión del videojuego, la fecha de revisión de la iteración será el día 30 de Marzo de 2013, en esta iteración se desarrollará las siguientes Historias de Usuario.

Tabla IV.LIII. Historias de Usuario a implementarse en la Iteración 1.

ID Historia de Usuario	Descripción
HU03	Como Jugador, quiero que los objetos aparezcan únicamente en la vía del tren, de modo que se puedan ir cogiendo.
HU07	Como Jugado, quiero que los controles del tren se encuentren en la parte inferior de la pantalla distribuidos en los extremos respectivamente, para que pueda facilitar la jugabilidad
HU08	Como Jugador, quiero que el tren pueda girar más de 90 grados con menos de 2 vagones, de modo que pueda darse la vuelta para cambiar el rumbo
HU09	Como Jugador, quiero que el tren no pueda girar más de 90 grados, de modo que si lo hiciera chocara con sus vagones y terminara el juego.
HU10	Como Jugador, quiero que el tren pueda moverse solo por los rieles o caminos, de modo que si se saliese de la vía termine el juego.
HU27	Como Sonidista, quiero que exista sonido cuando el tren este en movimiento, para dar un aspecto real al juego.

Elaborado por: Los Autores.

Tarjetas de Tarea de la Iteración 1

En la Reunión de Planificación de la Iteración 1 se centró específicamente en convertir las Historias de Usuario en Tareas y auto-asignar a los Miembros del Equipo DAV. A continuación en la **Figura IV.59.**, se muestra una de ellas, las demás Tarjetas de Tarea se encuentran en el **Anexo II.**

Tarjeta de Tarea	
Número de Tarea: T01	Historia de Usuario (Nro. Y Nombre): HU03-Programador
Nombre Tarea: <i>Los objetos aparezcan en la vía del tren.</i>	
Tipo de Tarea: Desarrollo	Puntos Estimados: 16 horas
Fecha Inicio: 26 de Marzo del 2013	Fecha Fin: 27 de Marzo del 2013
Miembro Responsable: Milton Saltos	
Descripción: <i>Los objetos deben ir apareciendo en los rieles por donde va a circular el tren para que los vayan cogiendo.</i>	

Figura IV.59. Tarjeta de Tarea “T01”.

Fuente: Los Autores.

Reserva de Iteración 1

Al finalizar la Iteración 1 todas las Tareas fueron completadas como se muestra en la **Figura IV.60.**, todos los días al finalizar las Reuniones Diarias se actualizó la Reserva de Iteración1, la cual fue siempre visible para todos los Miembros del Equipo DAV permitiendo ver como se fue avanzando.

ITERACIÓN	INICIA	TERMINA	DURACIÓN					
1	25 de Marzo de 2013	29 de Marzo de 2013	5 días					
Día Tareas Pendientes Horas de trabajo pendientes			M	M	J	V	S	
			26-mar	27-mar	28-mar	29-mar	30-mar	
			6	6	4	2	0	
			64	48	32	16	0	
Requisito	Tarea	Responsable	Estado	Esfuerzo				
N/A	T01	Milton Saltos	Terminada	16	8			
N/A	T02	Gabriel Amendániz	Terminada	16	8			
T01,T02	T03	Milton Saltos	Terminada	8	8	8		
T03	T04	Gabriel Amendániz	Terminada	8	8	8		
T01	T05	Milton Saltos	Terminada	8	8	8	8	
T01	T06	Gabriel Amendániz	Terminada	8	8	8	8	0

Figura IV.60. Reserva de Iteración 1.

Fuente: Los Autores.

En la **Figura IV.61.**, se observa cómo fue avanzando el desarrollo del videojuego a lo largo de la iteración en relación a las tareas que se fueron completando.

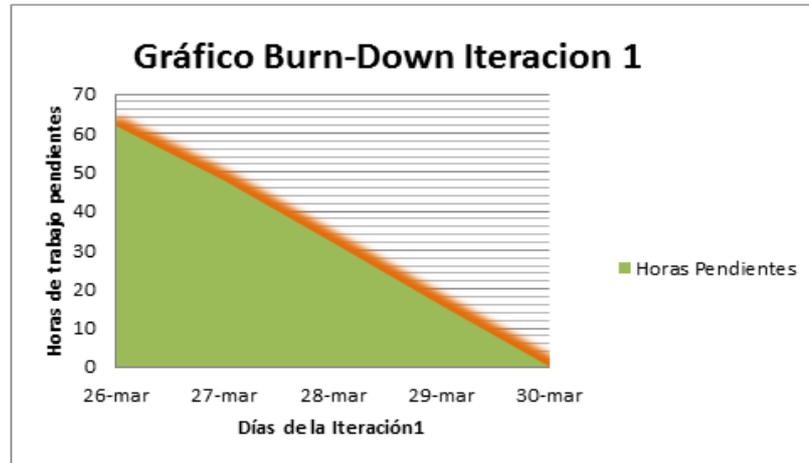


Figura IV.61. Gráfico Burn-Down de la Iteración 1.
Fuente: Los Autores.

En la gráfica se puede observar que la disminución de las horas fue constante de acuerdo a la pendiente que presenta, esto se debe a que las tareas realizadas fueron desarrolladas con éxito para la primera iteración del videojuego.

En la siguiente figura se puede observar el avance de las tareas realizadas a lo largo de la iteración en contra del tiempo.

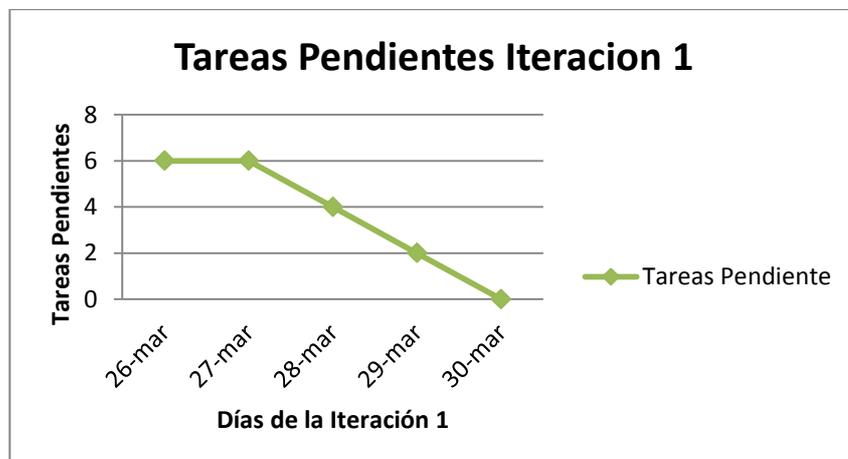


Figura IV.62. Tareas Pendientes de la Iteración 1.
Fuente: Los Autores.

El gráfico muestra que la disminución de las tareas fue constante a excepción del primer día debido a que la estimación supero las 8 horas de trabajo.

La práctica de Refactorización de Código se la llevo a cabo a lo largo del proyecto como se muestra en la **Figura IV.63.**, con el objetivo de evitar duplicaciones y mejorar la legibilidad para los Miembros del Equipo DAV y de esta manera hacerlo flexible para facilitar posteriores cambios.

```
public TextView textto;

// Variables.
private String ruta_imagen; // la ruta de la imagen que el usuario eligio
                          // para la imagen de su persona.
private int SELECCIONAR_IMAGEN = 237487;

// Constantes privadas.
private int CODIGO_RESULT_EDITAR_PERSONA = 0;
public TextView tvDatosRecibidos; //TextView donde se mostrara la cadena recibida
public String cadenaObtenida; //String donde se guardará la cadena recibida "PUNTUACIÓN"
public int cadenaObtenida1; //Int donde se guardará la cadena recibida "PEGUNTAS CORRECTAS"
public int cadenaObtenida2; //Int donde se guardará la cadena recibida "PEGUNTAS INCORRECTAS"

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.editar_persona);

    // Hace referencia a los objetos xml.
    botonGuardar = (Button) findViewById(R.id.botonGuardar);
    botonLimpiar = (Button) findViewById(R.id.botonLimpiar);
    editTextNombre = (EditText) findViewById(R.id.editTextNombre);
    editTextApellido = (EditText) findViewById(R.id.editTextApellido);
    editTextEdad = (EditText) findViewById(R.id.editTextEdad);
    botonImagenPersona = (Button) findViewById(R.id.botonAgregarImagenPersona);
    imagenPersona = (ImageView) findViewById(R.id.imagenPersona);
}
```

Figura IV.63. Refactorización del Código del videojuego “El Tren del Conocimiento”.

Fuente: Los Autores.

Reunión de Revisión de la Iteración 1

Esta reunión se la realizó el día 30 de Marzo de 2013 como estuvo previsto, se presentó la funcionalidad de la primera versión terminada, la demostración del videojuego se realizó en el entorno del jugador como se muestra en la siguiente figura.



Figura IV.64. Entrega Funcional de la Iteración 1.
Fuente: Los Autores.

Conclusión de la Reunión de Revisión de la Iteración 1

Todos los objetivos de la Iteración 1 fueron alcanzados con éxito, las tareas estuvieron bien distribuidas sin sobrecarga de trabajo para el Equipo DAV, de esta revisión se obtuvieron nuevas Historias de Usuario, las mismas ya se encuentran incluidas en la **Tabla IV.LI.**

Reunión de Retrospectiva

En esta reunión se analizó las prácticas, artefactos y herramientas utilizadas en el proyecto, decidiendo seguir con las mismas debido a que fueron excelentes para cumplir con éxito la Iteración 1.

Después de realizar las Reuniones de Revisión y Retrospectiva se dio por concluido la Iteración 1.

Iteración 2

El objetivo de esta iteración es obtener la segunda versión del videojuego, la fecha de revisión de la iteración será el día 06 de Abril de 2013, en esta iteración se desarrollará las siguientes Historias de Usuario.

Tabla IV.LIV. Historias de Usuario a implementarse en la Iteración 2.

ID Historia de Usuario	Descripción
HU02	Como Jugador, quiero que se muestre un splash al iniciar la aplicación (videojuego), de modo que sea divertido.
HU14	Como Jugador, quiero visualizar el puntaje al coger los objetos y al responder correctamente las preguntas, de modo que pueda conocer la puntuación que voy alcanzando.
HU15	Como Jugador, quiero pausar el juego al pulsar un botón, de modo que pueda hacer algo en ese momento o salir del juego.
HU19	Como Jugador, quiero que antes de empezar a jugar se visualice el nivel del juego en una pantalla de preparado, para que sea más óptimo.
HU20	Como Jugador, quiero pulsar en cualquier parte de la pantalla preparado, para empezar a jugar
HU28	Como Diseñador, quiero que el tren tenga un aspecto real, de modo que se muestre más divertido.
HU30	Como Diseñador, quiero que los objetos tengan un aspecto de cofres, hongos y monedas de colores dorados y cafés, de modo que se muestren divertidos.

Elaborado por: Los Autores.

Tarjetas de Tarea de la Iteración 2

En la Reunión de Planificación de la Iteración 2 de igual manera se centró específicamente en convertir las Historias de Usuario en Tareas y auto-asignar a los Miembros del Equipo DAV. En la **Figura IV.65.**, se muestra una de ellas, las demás Tarjetas de Tarea se encuentran en el **Anexo II.**

Tarjeta de Tarea	
Número de Tarea: T07	Historia de Usuario (Nro. Y Nombre): HU02-Programador
Nombre Tarea: Mostrar un Splash al iniciar la aplicación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 16 horas
Fecha Inicio: 02 de Abril del 2013	Fecha Fin: 03 de Abril del 2013
Miembro Responsable: Milton Saltos	
Descripción: Cuando se inicie el videojuego se debe mostrar un <i>Splash</i> para hacerlo más divertido.	

Figura IV.65. Tarjeta de Tarea “T07”.
Fuente: Los Autores.

Reserva de Iteración 2

Al finalizar la Iteración 2 todas las Tareas fueron completadas como se muestra en la **Figura IV.66.**, de igual manera todos los días al finalizar las Reuniones Diarias se actualizó la Reserva de Iteración 2, siendo esta siempre visible para todos los Miembros del Equipo DAV.

ITERACIÓN	INICIA	TERMINA	DURACIÓN					
2	01 de Abril de 2013	05 de Abril de 2013	5 días					
Día Tareas Pendientes Horas de trabajo pendientes			M	M	J	V	S	
			02-abri	03-abri	04-abri	05-abri	06-abri	
			7	7	5	3	0	
			64	48	32	16	0	
Requisito	Tarea	Responsable	Estado	Esfuerzo				
N/A	T07	Milton Saltos	Terminada	16	8			
T1	T08	Gabriel Amendáriz	Terminada	16	8			
T01	T09	Gabriel Amendáriz	Terminada	8	8	8		
T01	T10	Milton Saltos	Terminada	8	8	8		
T10	T11	Gabriel Amendáriz	Terminada	4	4	4	4	
N/A	T12	Milton Saltos	Terminada	8	8	8	8	
N/A	T13	Gabriel Amendáriz	Terminada	4	4	4	4	0

Figura IV.66. Reserva de Iteración 2.
Fuente: Los Autores.

En la **Figura IV.67.**, se observa cómo fue avanzando el desarrollo del videojuego a lo largo de la iteración en relación a las tareas que se fueron completando.

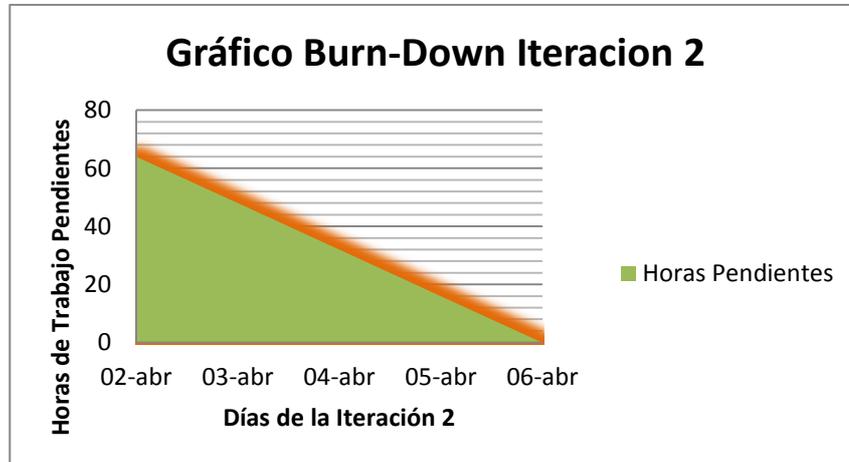


Figura IV.67. Gráfico Burn-Down de la Iteración 2.
Fuente: Los Autores.

En la gráfica se puede observar que la disminución de las horas fue constante de acuerdo a la pendiente que presenta igual que la Iteración 1, debido a que las tareas realizadas fueron desarrolladas con éxito.

En la siguiente figura se puede observar el avance de las tareas realizadas a lo largo de la iteración en contra del tiempo.

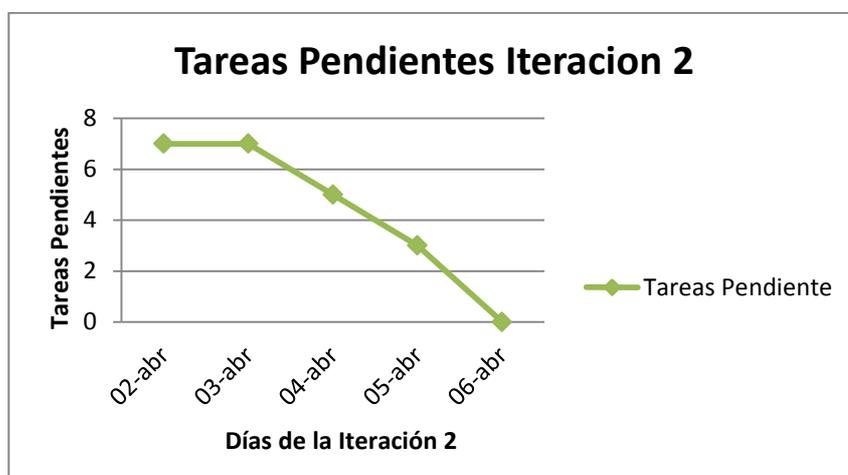


Figura IV.68. Tareas Pendientes de la Iteración 2.
Fuente: Los Autores.

El gráfico muestra que la disminución de las tareas fue constante a excepción del primer día esto se debe a que la estimación realizada para las dos Tareas (T02, T14) fue de 16 horas de trabajo.

Reunión de Revisión de la Iteración 2

Esta reunión se la realizó el día 06 de Abril de 2013 como estuvo previsto, se presentó la funcionalidad terminada de la segunda versión, la demostración fue presentada en el entorno del jugador como se muestra en la siguiente figura.



Figura IV.69. Entrega Funcional de la Iteración 2.
Fuente: Los Autores.

Conclusión de la Reunión de Revisión de la Iteración 2

Todos los objetivos de la Iteración 2 fueron alcanzados con éxito, las tareas estuvieron bien distribuidas al igual que la Iteración 1 sin sobrecargar el trabajo al Equipo DAV.

Reunión de Retrospectiva

En esta reunión se estableció seguir en todas las demás iteraciones con las prácticas, artefactos y herramientas utilizadas debido a que fueron excelentes para cumplir con éxito la Iteración 2.

Después de realizar las Reuniones de Revisión y Retrospectiva se dio por concluido la Iteración 2.

Iteración 3

El objetivo de esta iteración es obtener la tercera versión del videojuego, la fecha de revisión de la iteración será el día 13 de Abril de 2013, en esta iteración se desarrollará las siguientes Historias de Usuario.

Tabla N° IV.LV. Historias de Usuario a implementarse en la Iteración 3.

ID Historia de Usuario	Descripción
HU01	Como Jugador, quiero tener un menú principal de modo que pueda ingresar a jugar, visualizar las instrucciones, créditos, puntuaciones y salir de videojuego.
HU05	Como Jugador, quiero visualizar los créditos del juego, para saber quién lo desarrollo.
HU24	Como Jugador, quiero que al escoger en el menú principal la opción salir se muestre una pantalla de confirmación de cerrar la aplicación., de modo que si no estoy seguro de salir pueda regresar al menú principal o simplemente cerrar la aplicación.
HU26	Como Jugador, quiero que exista un botón de sonido en la pantalla de menú principal, para poder habilitar y deshabilitar el sonido.
HU29	Como Diseñador, quiero que los vagones tengan un aspecto de un cuadrado de color rojo, de modo que se muestre realista.
HU33	Como Sonidista, quiero que exista audio de fondo en las diferentes pantallas del videojuego, de modo que sea divertido.
HU04	Como Jugador quiero visualizar las instrucciones del juego, para saber cómo jugar
HU11	Como Jugador, quiero que el tren al coger los objetos se escuche un sonido de “incremento”, para poder dar realismo al juego.

Elaborado por: Los Autores.

Tarjetas de Tarea de la Iteración 3

En la Reunión de Planificación de la Iteración 3 se centró de igual manera en convertir las Historias de Usuario en Tareas para asignar a los Miembros del Equipo DAV. A continuación en la **Figura IV.70.**, se muestra una de ellas, las demás Tarjetas de Tarea se encuentran en el **Anexo II.**

Tarjeta de Tarea	
Número de Tarea: T14	Historia de Usuario (Nro. Y Nombre): HU01-Programador
Nombre Tarea: Realizar menú principal	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8 horas
Fecha Inicio: 09 de Abril de 2013	Fecha Fin: 09 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: <i>Realizar el menú principal del videojuego de modo el jugador pueda ingresar a jugar, visualizar las Instrucciones, créditos, Puntuaciones y pueda salir de la aplicación (videojuego).</i>	

Figura IV.70. Tarjeta de Tarea “T14”.

Fuente: Los Autores.

Reserva de Iteración 3

Al finalizar la Iteración 3 todas las Tareas fueron completadas como se muestra en la **Figura IV.71.**, todos los días de la misma manera al finalizar las Reuniones Diarias se actualizó la Reserva de Iteración 3, siendo esta siempre visible para todos los Miembros.

ITERACIÓN	INICIA	TERMINA	DURACIÓN					
3	08 de Abril de 2013	12 de Abril de 2013	5 días					
Dia Tareas Pendientes Horas de trabajo pendientes			M	M	J	V	S	
			09-abri	10-abri	11-abri	12-abri	13-abri	
			8	6	4	2	0	
			64	48	32	16	0	
Requisito	Tarea	Responsable	Estado	Esfuerzo				
N/A	T14	Gabriel Amendáriz	Terminada	8				
N/A	T15	Milton Saltos	Terminada	8				
T14	T16	Milton Saltos	Terminada	4	4			
T14	T17	Gabriel Amendáriz	Terminada	16	16	8		
N/A	T18	Milton Saltos	Terminada	4	4	0		
T14	T19	Milton Saltos	Terminada	8	8	8		
N/A	T20	Gabriel Amendáriz	Terminada	8	8	8	8	
T01	T21	Milton Saltos	Terminada	8	8	8	8	0

Figura IV.71. Reserva de Iteración 3.

Fuente: Los Autores.

En la **Figura IV.72.**, se observa cómo fue avanzando el desarrollo del videojuego a lo largo de la iteración en relación a las tareas que fueron completadas.

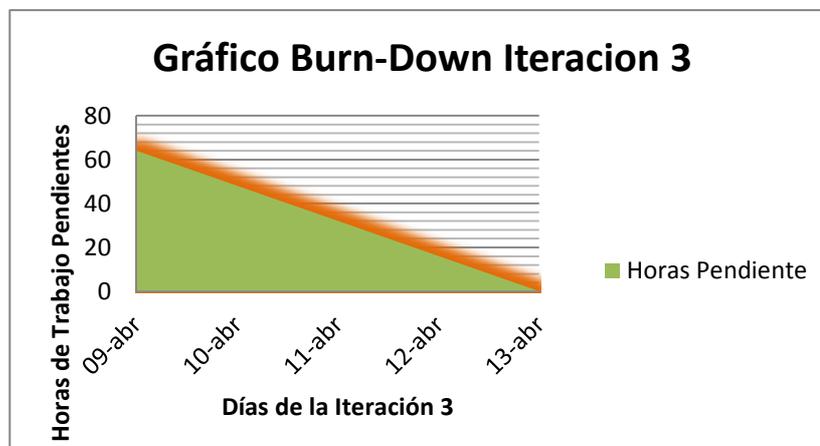


Figura IV.72. Gráfico Burn-Down de la Iteración 3.

Fuente: Los Autores.

En la gráfica se puede observar que la disminución de las horas fue constante de acuerdo a la pendiente que presenta, completándose las Tareas con normalidad con buenos tiempos.

En cambio en la **Figura IV.73.**, se puede observar el avance de las tareas realizadas a lo largo de la iteración en contra del tiempo.

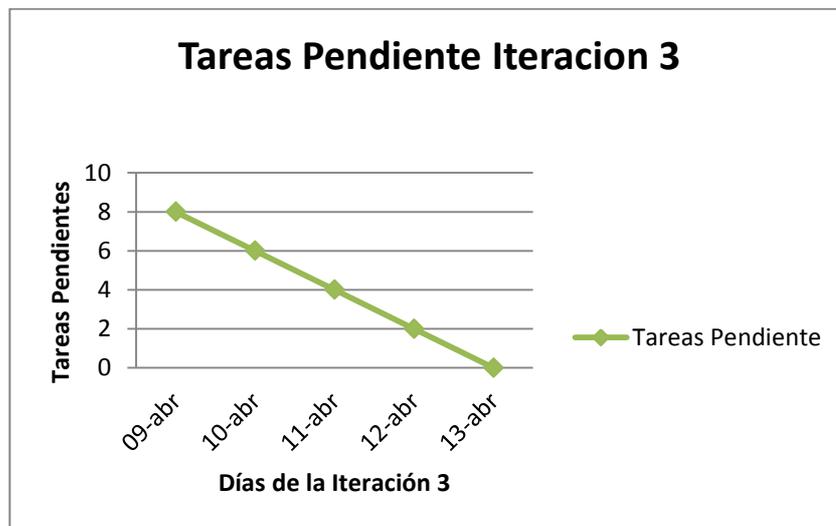


Figura IV.73. Tareas Pendientes de la Iteración 3.
Fuente: Los Autores.

El gráfico muestra que la disminución de las tareas fue constante debido a que la estimación de las horas fue buena.

Reunión de Revisión de la Iteración 3

Esta reunión se la realizó el día 13 de Abril de 2013 como estuvo previsto, se presentó la funcionalidad terminada de la tercera versión, la cual fue demostrada en el entorno del jugador como se muestra en la **Figura IV.74.**



Figura IV.74. Entrega Funcional de la Iteración 3.
Fuente: Los Autores.

Conclusión de la Reunión de Revisión de la Iteración 3

Todos los objetivos de la Iteración 3 fueron alcanzados con éxito, sin existir sobrecarga de trabajo al Equipo DAV.

Después de realizar las Reuniones de Revisión se dio por concluido la Iteración 3.

Iteración 4

El objetivo de esta iteración es obtener la cuarta versión del videojuego, la fecha de revisión de la iteración será el día 20 de Abril de 2013, en esta iteración se desarrollará las siguientes Historias de Usuario.

Tabla IV.LVI. Historias de Usuario a implementarse en la Iteración 4.

ID Historia de Usuario	Descripción
HU06	Como Jugador, quiero visualizar las 10 primeras altas puntuaciones, de modo que pueda saber cuáles jugadores la obtuvieron.
HU18	Como Jugador, quiero que las preguntas vayan apareciendo a medida

	que vaya cogiendo 6 objetos, para aumentar la diversión del juego.
HU21	Como Jugador, quiero visualizar una pantalla de “Fin del Juego” al perder el juego conjuntamente con dos botones de “Nuevo Juego” y “Salir”, para saber que perdí y poder intentarlo de nuevo o salir del juego
HU22	Como Jugador, quiero poder responder las preguntas que se van mostrando mediante dos botones de afirmación (SI) o negación (NO), para poder saber el grado de conocimiento que tengo sobre la Provincia de Chimborazo y adquirir nuevo conocimiento.
HU35	Como Diseñador, quiero que exista en la pantalla de Instrucciones botones de navegabilidad entre sus pantallas, para poder navegar fácilmente.
HU32	Como Diseñador, quiero que exista en la pantalla de créditos, instrucciones, puntuaciones, niveles y salir un botón de regreso, de modo que pueda volver al menú principal.
HU25	Como Jugador, quiero que existan 2 botones para poder acceder a las redes sociales del Facebook o Twitter, para conocer opiniones del juego e información adicional del mismo.

Elaborado por: Los Autores.

Tarjetas de Tarea de la Iteración 4

En la Reunión de Planificación de la Iteración 4 se centró específicamente en convertir las Historias de Usuario en Tareas para asignar a los Miembros del Equipo DAV. A continuación en la **Figura IV.75.**, se muestra una de ellas, las demás Tarjetas de Tarea se encuentran en el **Anexo II.**

Tarjeta de Tarea	
Número de Tarea: T22	Historia de Usuario (Nro. Y Nombre): HU06-Programador
Nombre Tarea: Visualizar las 10 primeras altas puntuaciones	
Tipo de Tarea: Desarrollo	Puntos Estimados: 16 horas
Fecha Inicio: 16 de Abril de 2013	Fecha Fin: 17 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Visualizar las 10 más altas puntuaciones de los jugadores en una lista.	

Figura IV.75. Tarjeta de Tarea “T22”.

Fuente: Los Autores.

Reserva de Iteración 4

Al finalizar la Iteración 4 todas las Tareas fueron completadas como se muestra en la **Figura IV.76.**, todos los días al finalizar las Reuniones Diarias se actualizó la Reserva de Iteración 4, la cual fue siempre visible para todos.

ITERACIÓN	INICIA	TERMINA	DURACIÓN					
4	15 de Abril de 2013	19 de Abril de 2013	5 días					
Día			M	M	J	V	S	
			16-abri	17-abri	18-abri	19-abri	20-abri	
			Tareas Pendientes	7	6	5	2	0
Horas de trabajo pendientes			64	48	34	16	0	
Requisito	Tarea	Responsable	Estado	Esfuerzo				
T24	T22	Gabriel Armendáriz	Terminada	16	8			
T01	T23	Milton Saltos	Terminada	8	0			
T05	T24	Milton Saltos	Terminada	8	8	2		
T22, T23, T24	T25	Gabriel Armendáriz	Terminada	8	8	8		
T20	T26	Milton Saltos	Terminada	8	8	8	0	
T15	T27	Gabriel Armendáriz	Terminada	8	8	8	8	
NT14	T28	Milton Saltos	Terminada	8	8	8	8	0

Figura IV.76. Reserva de Iteración 4.

Fuente: Los Autores.

En la **Figura IV.77.**, se observa cómo fue avanzando el desarrollo del videojuego a lo largo de la iteración en relación a las tareas que se fueron completando.

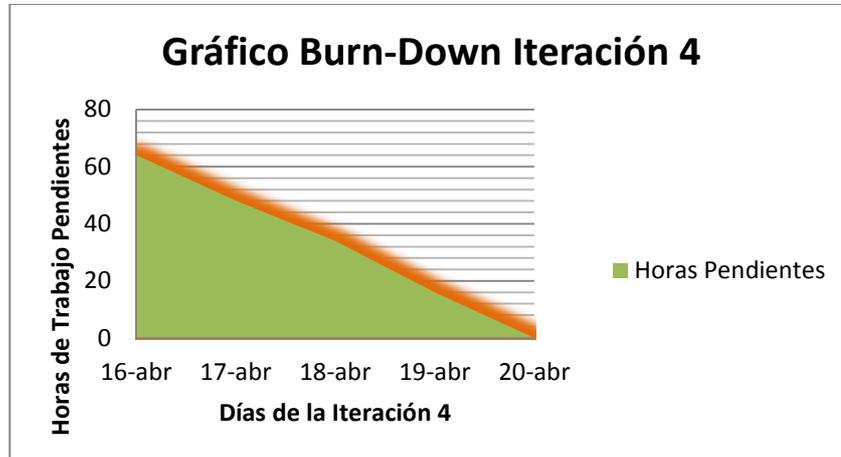


Figura IV.77. Gráfico Burn-Down de la Iteración 4.
Fuente: Los Autores.

En la gráfica se puede observar que la disminución de las horas fue constante a excepción en el desarrollo de una tarea debido a que no se pudo terminarla durante el día establecido, pero fue solucionado en el día siguiente acumulando una sobrecarga de 2 horas, completando así con éxito todas las tareas.

En la siguiente imagen se puede observar el avance de las tareas realizadas a lo largo de la iteración en contra del tiempo.

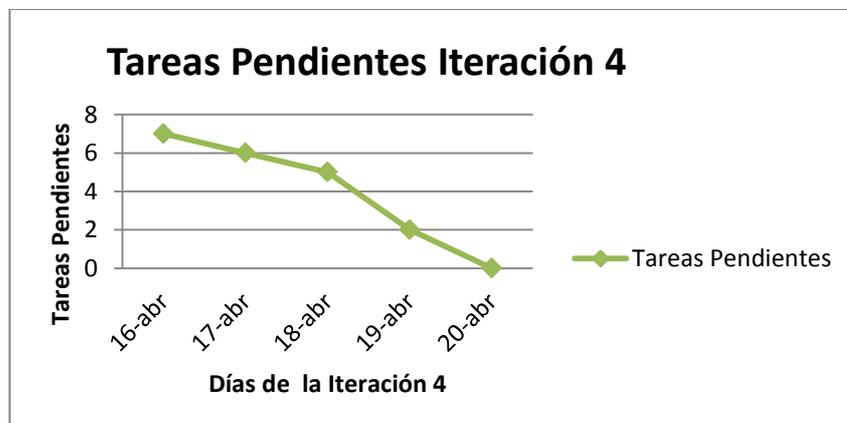


Figura IV.78. Tareas Pendientes de la Iteración 4.
Fuente: Los Autores.

El gráfico muestra que la disminución de las tareas fue constante a excepción del inconveniente con una de las tareas; de esta manera se culminó con éxito todo lo establecido para la iteración.

Reunión de Revisión de la Iteración 4

Esta reunión se la realizó el día 20 de Abril de 2013 como estuvo previsto, se presentó la funcionalidad de la cuarta versión, en donde se realizó la demostración del videojuego en el entorno del jugador como se muestra en la **Figura IV.79**.



Figura IV.79. Entrega Funcional de la Iteración 4.

Fuente: Los Autores.

Conclusión de la Reunión de Revisión de la Iteración 4

Todos los objetivos de la Iteración 4 fueron alcanzados con éxito a pesar del inconveniente con una tarea que no fue estimada con su valor real..

Después de realizar las Reuniones de Revisión se dio por concluido la Iteración 5.

Iteración 5

El objetivo de esta iteración es obtener la quinta versión del videojuego, la fecha de revisión de la iteración será el día 27 de Abril de 2013, en esta iteración se desarrollará las siguientes Historias de Usuario.

Tabla IV.LVII. Historias de Usuario a implementarse en la Iteración 5.

ID Historia de Usuario	Descripción
HU12	Como Jugador, quiero que existan 8 niveles los mismos que se encuentren deshabilitados exceptuando el primero, de modo que al ir completando un nivel se vaya deshabilitando el siguiente.
HU31	Como Diseñador, quiero que existan animaciones en la pantalla del menú principal, créditos, instrucciones, niveles y salir, de modo que den un aspecto divertido
HU13	Como Jugador, quiero Ingresar mi nombre y apellido y una foto si lo deseara, para poder visualizar junto a la puntuación alcanzada.
HU16	Como Jugador, quiero cancelar el ingreso de datos, para no visualizar la puntuación.
HU23	Como Jugador, quiero que en cada nivel se muestren como máximo 5 preguntas, para que el juego no sea aburrido.
HU17	Como Jugador, quiero que el tren aumente los vagones a medida que vaya cogiendo 3 objetos, para aumentar la diversión del juego.
HU34	Como Diseñador, quiero que la pantalla de puntuaciones muestre el nombre, apellido, puntuación, número de preguntas correctas e incorrectas contestadas de los 10 jugadores con más altas puntuaciones, de modo que se pueda apreciar mejor los datos de los jugadores
HU37	Como Jugador, quiero que el contenido de las preguntas sea sobre la provincia de Chimborazo, de modo que pueda complementar mi conocimiento.

Elaborado por: Los Autores.

Tarjetas de Tarea de la Iteración 5

En la Reunión de Planificación de esta iteración se centró de igual manera en convertir las Historias de Usuario en Tareas y asignar a los Miembros del Equipo DAV. A continuación en la **Figura IV.80.**, se muestra una de ellas, las demás Tarjetas de Tarea se encuentran en el **Anexo II.**

Tarjeta de Tarea	
<i>Número de Tarea:</i> T29	<i>Historia de Usuario (Nro. Y Nombre):</i> HU12-Programador
<i>Nombre Tarea:</i> Existen 8 niveles deshabilitados a excepción del primero	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 16 horas
<i>Fecha Inicio:</i> 23 de Abril de 2013	<i>Fecha Fin:</i> 24 de Abril de 2013
<i>Miembro Responsable:</i> Gabriel Armendáriz	
<i>Descripción:</i> Deben existir 8 niveles los mismos que se encuentren deshabilitados exceptuando el primero, de modo que al ir completando un nivel se vaya deshabilitando el siguiente.	

Figura IV.80. Tarjeta de Tarea “T29”.
Fuente: Los Autores.

Reserva de Iteración 5

Al finalizar la Iteración 5 todas las Tareas fueron completadas como se muestra en la **Figura IV.81.**, todos los días al finalizar las Reuniones Diarias se actualizaba la Reserva de Iteración1, la cual fue siempre visible para todos los Miembros del Equipo DAV permitiendo ver como se fue avanzando.

ITERACIÓN	INICIA	TERMINA	DURACIÓN					
5	22 de Abril de 2013	26 de Abril de 2013	5 días					
Día Tareas Pendientes Horas de trabajo pendientes			M	M	J	V	S	
			23-abr	24-abr	25-abr	26-abr	27-abr	
			8	8	6	3	0	
			64	48	32	16	0	
Requisito	Tarea	Responsable	Estado	Esfuerzo				
T01, T02, T03, T04, T05, T06	T29	Gabriel Armendáriz	Terminada	16	8			
T14, T15, T16, T17, T18, T19, T20, T21	T30	Milton Saltos	Terminada	16	8			
T22	T31	Gabriel Armendáriz	Terminada	8	8	8		
T30	T32	Milton Saltos	Terminada	4	4	4		
T29	T33	Milton Saltos	Terminada	4	4	4		
T29	T34	Gabriel Armendáriz	Terminada	4	4	4	4	
T31	T35	Gabriel Armendáriz	Terminada	4	4	4	4	
N/A	T36	Milton Saltos	Terminada	8	8	8	8	0

Figura IV.81. Reserva de Iteración 4.

Fuente: Los Autores.

En la siguiente figura se observa cómo fue avanzando el desarrollo del videojuego a lo largo de la iteración en relación a las tareas que se fueron completando.

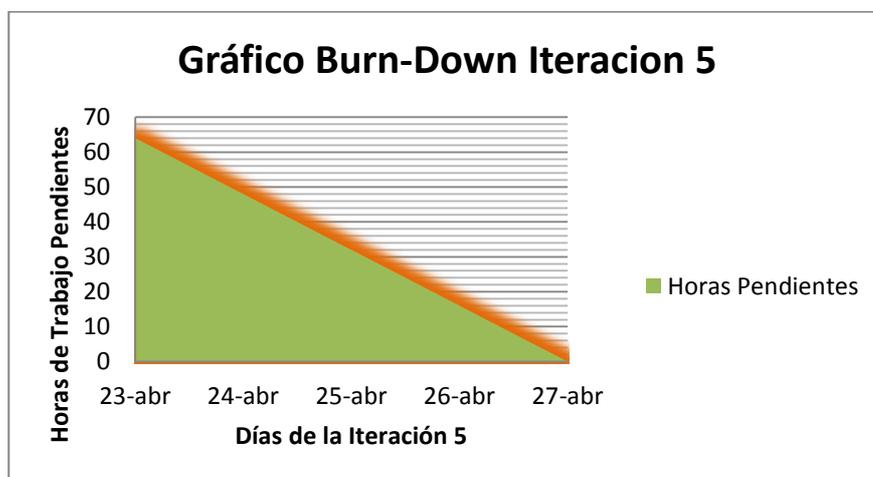


Figura IV.82. Gráfico Burn-Down de la Iteración 5.

Fuente: Los Autores

En la gráfica se puede observar que la disminución de las horas fue constante de acuerdo a la pendiente que presenta, esto se debe a que las tareas realizadas fueron desarrolladas sin ningún inconveniente en los tiempos establecidos.

En la siguiente figura se puede observar el avance de las tareas realizadas a lo largo de la iteración en contra del tiempo.

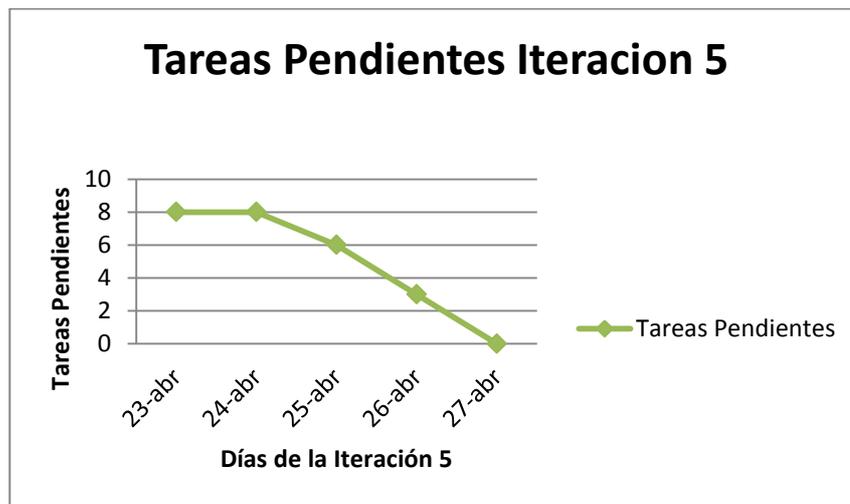


Figura IV.83. Tareas Pendientes de la Iteración 5.

Fuente: Los Autores.

El gráfico muestra que la disminución de las tareas fue constante a excepción del primer día debido a que las horas estimadas fueron de 16 horas superando el día de trabajo.

Reunión de Revisión de la Iteración 5

Esta reunión se la realizó el día 28 de Abril de 2013 como estuvo previsto, se presentó la funcionalidad de la quinta iteración, en donde se realizó la demostración del videojuego en el entorno del jugador como se muestra en la **Figura IV.84**.



Figura IV.84. Entrega Funcional de la Iteración 5.

Fuente: Los Autores.

Conclusión de la Reunión de Revisión de la Iteración 5

Todos los objetivos de la Iteración 5 fueron alcanzados con éxito, debido a que no existió sobrecarga de trabajo.

Después de realizar las Reuniones de Revisión se dio por concluido la Iteración 5.

Iteración 6

El objetivo de esta iteración es obtener la sexta y última versión del videojuego, la fecha de revisión de la iteración será el día 04 de Mayo de 2013, en esta iteración se desarrollará las últimas Historias de Usuario que se muestra a continuación.

Tabla IV.LVIII. Historias de Usuario a implementarse en la Iteración 6.

ID Historia de Usuario	Descripción
HU38	Como Diseñador, quiero que al responderse una pregunta correctamente se muestre un mensaje de “CORRECTO” con un visto de color verde, con un botón de “continuar” para seguir jugando, conjuntamente con un sonido de correcto., créditos, instrucciones, niveles, y salir del juego, de modo que pueda saber que se respondió

	correctamente.
HU39	Como Diseñador, quiero que al responderse una pregunta incorrectamente se muestre un mensaje de “INCORRECTO” con una x de color rojo, un botón de “continuar” para seguir jugando, conjuntamente con un sonido de incorrecto, de modo que pueda saber que se respondió incorrectamente.
HU40	Como Jugador, quiero que al responder una pregunta incorrectamente se muestre la respuesta correcta, de modo pueda saber que es lo correcto.
HU41	Como Jugador, quiero la puntuación del juego se vaya acumulando nivel tras nivel, para conocer el total alcanzado al finalizar el juego.
HU42	Como Jugador, quiero que la puntuación disminuya cuando se responda incorrectamente en 10 puntos, de modo que no pueda alcanzar la máxima puntuación
HU43	Como Jugador, quiero que al completar todos los niveles y al responder correctamente todas las preguntas se muestre una pantalla de congratulación, de modo que pueda sentirme elogiado.
HU36	Como Artista, quiero que se escuche un sonido al iniciarse el Splash, de modo que sea divertido.
HU44	Como Jugador, quiero que al completar un nivel correctamente se muestre una pantalla de nivel completado, con un botón de “CONTINUAR” y otro de “SALIR”, de modo que pueda saber que complete el nivel correctamente.

Elaborado por: Los Autores.

Tarjetas de Tarea de la Iteración 6

En la Reunión de Planificación de la Iteración 6 se centró específicamente en convertir las Historias de Usuario en Tareas para asignar a los Miembros del Equipo DAV. A continuación en la **Figura IV.85.**, se muestra una de ellas, las demás Tarjetas de Tarea se encuentran en el **Anexo II.**

Tarjeta de Tarea	
Número de Tarea: T37	Historia de Usuario (Nro. Y Nombre): HU38-Diseñador
Nombre Tarea: Responder preguntas correctamente	
Tipo de Tarea: Diseño	Puntos Estimados: 8 horas
Fecha Inicio: 30 de Abril de 2013	Fecha Fin: 30 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Cuando se respondo una pregunta correctamente se muestre un mensaje de "CORRECTO" con un visto de color verde, con un botón de "continuar" para seguir jugando, conjuntamente con un sonido de correcto en créditos, instrucciones, niveles, y salir del juego, de modo que pueda saber que se respondió correctamente.	

Figura IV.85. Tarjeta de Tarea "T37".

Fuente: Los Autores.

Reserva de Iteración 6

Al finalizar la Iteración 6 todas las Tareas fueron completadas como se muestra en la Figura IV.86., todos los días al finalizar las Reuniones Diarias se actualizó la Reserva de la Iteración 6, la cual fue siempre visible para todos.

ITERACIÓN	INICIA	TERMINA	DURACIÓN					
6	29 de Abril de 2013	03 de Mayo de 2013	5 días					
Dia Tareas Pendientes Horas de trabajo pendientes			M	M	J	V	S	
			30-abri	01-mayo	02-mayo	03-mayo	04-mayo	
			8	6	3	0	0	
			40	24	8	0	0	
Requisito	Tarea	Responsable	Estado	Esfuerzo				
T23	T37	Gabriel Armendáriz	Terminada	8	0			
T23	T38	Milton Saltos	Terminada	8	0			
T38	T39	Gabriel Armendáriz	Terminada	8	8	0		
T31	T40	Milton Saltos	Terminada	4	4	0		
T31, T38	T41	Milton Saltos	Terminada	4	4	0		
T29, T37, T23	T42	Gabriel Armendáriz	Terminada	4	4	4	0	
T07	T43	Milton Saltos	Terminada	2	2	2	0	
T29, T37, T23	T44	Milton Saltos	Terminada	2	2	2	0	0

Figura IV.86. Reserva de Iteración 6.

Fuente: Los Autores.

En la **Figura IV.87.**, se observa cómo fue avanzando el desarrollo del videojuego a lo largo de la última iteración en relación a las tareas que se fueron completando.

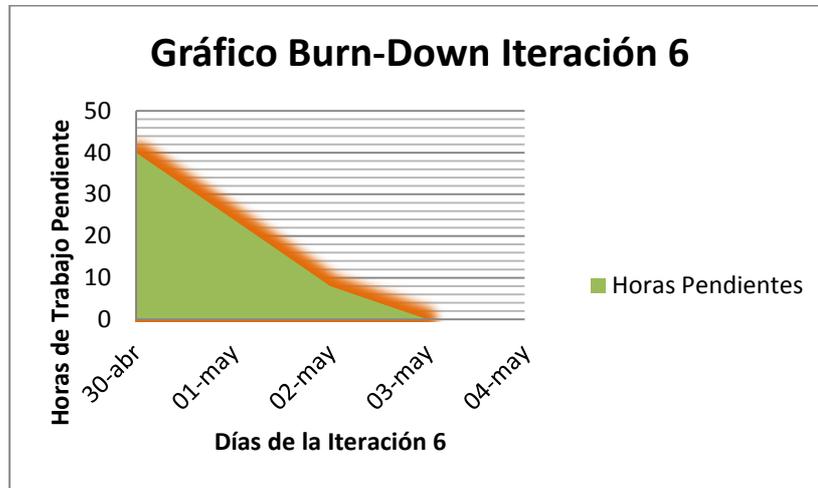


Figura IV.87. Gráfico Burn-Down de la Iteración 6.
Fuente: Los Autores.

En la gráfica se puede observar que la disminución de las horas fue constante sin ningún inconveniente, con el cumplimiento de estas tareas se termina con el desarrollo del videojuego.

En la siguiente figura se puede observar el avance de las tareas realizadas a lo largo de la última iteración en contra del tiempo.

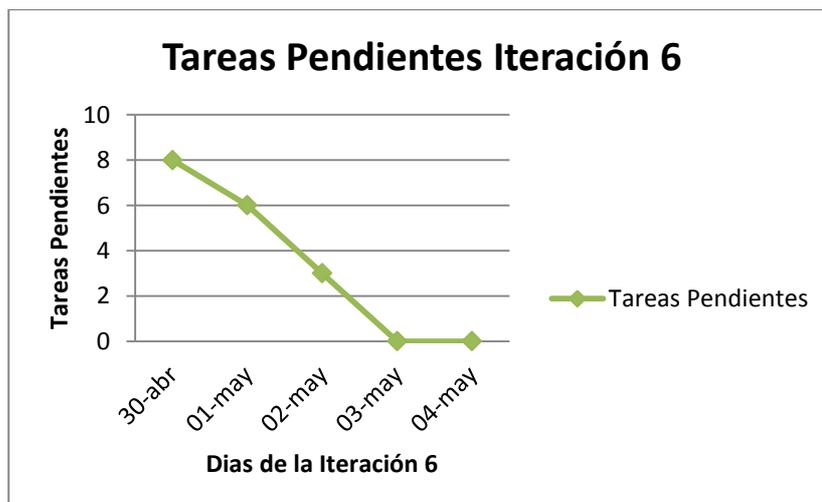


Figura IV.88. Tareas Pendientes de la Iteración 6.
Fuente: Los Autores.

El gráfico muestra que la disminución de las tareas fue constante sin existir inconvenientes ni sobrecarga de trabajo.

Reunión de Revisión de la Iteración 6

Esta reunión se la realizó el día 04 de Mayo de 2013 como estuvo previsto, se presentó la última funcionalidad en donde estuvo integrado todas las versiones obtenidas en las iteraciones anteriores, demostrando así el videojuego final en el entorno del jugador como se muestra en la siguiente figura:



Figura IV.89. Entrega Funcional de la Iteración 6.

Fuente: Los Autores.

Conclusión de la Reunión de Revisión de la Iteración 6

Todos los objetivos de la Iteración 6 fueron alcanzados sin ningún problema, las tareas estuvieron bien distribuidas sin existir sobrecarga de trabajo para el Equipo DAV, concluyendo así con todas las funcionalidades del videojuego.

4.3. FASE POST-JUEGO

Una vez desarrollado y completado con éxito todos los requerimientos funcionales del videojuego y sin existir más Historias de Usuario por parte del Dueño del Producto para ser implementadas se da por concluido y se cierra el proyecto.

Se libera el videojuego “EL TREN DEL CONOCIMIENTO” y se verifica la versión final.

4.4. COMPROBACIÓN DE LA HIPÓTESIS

4.4.1. Hipótesis Planteada

La adaptación de una metodología en el desarrollo de videojuegos, facilitará la creación de los juegos en la plataforma Android de forma técnica y sencilla.

4.4.2. Tipo de hipótesis

La Hipótesis de esta investigación es de tipo Causa - Efecto.

Causa: La adaptación de una metodología en el desarrollo de videojuegos.

Efecto: Facilita la creación de los juegos en la plataforma Android de forma técnica y sencilla.

4.4.3. Determinación de las variables

- **Variable Independiente:** Adaptación de una metodología.
- **Variable Dependiente:** Facilidad de creación de juegos en la plataforma Android.

4.4.4. Operacionalización Conceptual

Tabla IV.LIX. Operacionalización Conceptual.

VARIABLE	TIPO	CONCEPTO
Adaptación de una metodología en el desarrollo de videojuegos.	Variable Independiente	Creación de una metodología en base a los beneficios y fortalezas que brindan las metodologías ágiles Scrum y Extreme Game Development para desarrollar videojuegos.
Facilidad de creación de juegos en la plataforma Android.	Variable Dependiente	Desarrollo fácil de videojuegos de forma técnica y sencilla en la plataforma Android.

Elaborado por: Los Autores.

4.4.5. Operacionalización Metodológica

Tabla IV.LX. Operacionalización Metodológica.

VARIABLE	INDICADOR	TÉCNICA	INSTRUMENTOS
Adaptación de una metodología en el desarrollo de video juegos.	Roles	Observación, Revisión de documentos y Pruebas.	Internet, Prototipo
	Reuniones	Observación, Revisión de documentos y Pruebas.	Internet, Prototipo
	Prácticas	Observación Revisión de documentos.	Internet, Prototipo
	Valores	Observación, Revisión de documentos.	Internet
	Fases	Observación, Revisión de documentos y Pruebas.	Internet, Prototipo
	Actividades	Observación, Revisión de documentos y Pruebas.	Internet, Prototipo
	Artefactos	Observación, Revisión de documentos y Pruebas.	Internet, Prototipo
	Herramientas	Observación, Revisión de documentos y Pruebas.	Internet, Prototipo

Facilidad de creación de juegos en la plataforma Android.	Roles	Observación y Pruebas	Prototipo
	Reuniones	Observación y Pruebas	Prototipo
	Prácticas	Observación y Pruebas	Prototipo
	Valores	Observación y Pruebas	Interne, Prototipo
	Fases	Observación y Pruebas	Prototipo
	Actividades	Observación y Pruebas	Prototipo
	Artefactos	Observación y Pruebas	Prototipo
	Herramientas	Observación y Pruebas	Prototipo

Elaborado por: Los Autores.

4.4.6. Demostración de la Hipótesis

Para la comprobación de la hipótesis anteriormente citada se presenta a continuación los resultados obtenidos del estudio y análisis que proyectaron a través de la adaptación de la nueva metodología y su utilización en el desarrollo del videojuego “El Tren del Conocimiento”.

Tabla IV.LXI. Demostración de la Hipótesis.

INDICADOR	RESULTADOS OBTENIDOS
Roles	Las metodologías DAV y SCUM definen tres roles principales entre los cuales se dividen todas las responsabilidades del proyecto, siendo estos: el Dueño del Producto que es el encargado de encaminar las necesidades del videojuego, escuchando a las partes interesadas y transmitiéndolas al Equipo de Desarrollo, analizando las mejoras que éstas puedan otorgar un mayor valor, además ajusta las características del videojuego y responde por la rentabilidad del producto (ROI), administrando y priorizando la Reserva de Producto; DAV Master, es un auténtico servidor neutral, encargado de enseñar la metodología a cada integrante implicado en el proyecto, preocupándose en poner la metodología en práctica de modo que se encuentre dentro de la cultura de la organización, entregando las ventajas y facilitando el desarrollo; Equipo de Desarrollo, es un equipo multidisciplinario y auto-organizado que puede ser de 3 a 9 personas incluyendo un Tester en el caso de la metodología DAV, encargados de construir el videojuego que el Dueño del Producto especifica. Conteniendo así los roles suficientes y óptimos para equipos de trabajo pequeños, en el caso de XGD los roles principales son siete: Development Team, Manager, Customer, Tester, Tracker, Coach, Gestor, recomendables para equipos de trabajo grandes, conteniendo roles que pueden estar incluidos en uno solo.

Reuniones	<p>Las metodologías DAV y SCRUM sugieren reuniones diarias para conocer las tareas realizadas individualmente, mejorar la comunicación del equipo e identificar y eliminar obstáculos a tiempo; también sugieren reuniones al iniciar un ciclo que permite planificar las tareas a ser realizadas por cada miembro del equipo y al finalizar la iteración se debe realizar una nueva reunión para presentar la funcionalidad obtenida del videojuego al dueño del producto y así conseguir una retroalimentación que permita evolucionar y darle un valor real a la Reserva del Producto, seguida por una última reunión de retrospectiva con todos los miembros del Equipo, para analizar cómo ha ido avanzando la iteración desde todos los puntos de vista y tratar de mejorarlos para garantizar el avance continuo de los procesos y desarrollos. La metodología XGD también sugiere reuniones pero no se considera por seguir un proceso diferente a las dos metodologías anteriores debido a la cantidad de fases que posee.</p>
Prácticas	<p>Las prácticas de la metodología SCRUM se enfocan únicamente en la gestión del proyecto, no posee prácticas de ingeniería, promoviendo entregas frecuentes jugables, permitiendo al cliente ser parte del equipo de desarrollo, además promueve planes de riesgos y mitigaciones frecuentes así como la transparencia en la planificación por penalizar a ningún miembro por reconocer o describir problemas imprevistos. En el caso de la metodología XGD promueve prácticas de Ingeniería y Gestión del Proyecto permitiendo a los miembros del equipo trabajar en parejas y permanecer en comunicación constante, compartiendo todo el código fuente del proyecto y desarrollando las tareas lo más sencillo posible para producir versiones jugables en periodos de tiempo cortos; además se promueva la refactorización de código para evitar código duplicado mejorando la</p>

	<p>estructura interna sin alterar su comportamiento externo.</p> <p>La metodología DAV posee prácticas tanto de SCRUM como de XGD de este modo se enfoca en la ingeniería y gestión del proyecto permitiendo la creación de videojuegos de forma técnica y sencilla.</p>
Valores	<p>En lo referente a valores la metodología DAV promueve el comportamiento individual y grupal, mientras que la metodología SCRUM promueve el comportamiento individual y la metodología XGD promueve el comportamiento grupal. Al unir lo mejor de las 2 metodologías se consigue que los involucrados en el desarrollo del juego estén dispuestos a comprometerse en alcanzar la meta, puedan tomar decisiones propias y oportunas, manteniendo todo acerca del proyecto visible y respetando las formas de pensar de los demás individuos en las reuniones grupales. La comunicación oral es necesaria en las reuniones para disminuir la cantidad de documentos y permitir expresar sus puntos de vista referentes al desarrollo del proyecto y de las pruebas funcionales que se presenten del videojuego.</p>
	<p>Con respecto a las fases las metodologías DAV y SCRUM definen su trabajo en tres fases, suficientes y necesarias para desarrollar un videojuego de calidad, fáciles a seguir, abarcan toda la gestión del proyecto y sobretodo cumplen con un seguimiento estricto en la planificación, tareas, iteraciones y tiempos de entregas establecidos. Estas fases son: Pre-Juego, en donde se realiza la concepción de la idea del juego, es decir, los aspectos fundamentales que conformarán el videojuego (genero, historia, bocetos, aspecto, interfaz de usuario, objetivos, reglas, características, jugabilidad, diseño de niveles, requerimientos técnicos y marketing), además se define la arquitectura y se realiza un diseño conceptual de la solución; Juego, en esta fase que suele llamarse</p>

Fases	<p>Iteración ya se tiene bien definido el alcance del proyecto, por lo tanto ya existe una idea clara de lo que realmente es el videojuego y lo que se debe hacer, consta de ciclos que pueden durar de 1 a 4 semanas donde se desarrollan las funcionalidades del videojuego; Post-juego, esta fase comienza cuando el equipo ha desarrollado con éxito todas las Historias de Usuario y el Dueño del Producto ya no tiene más funcionalidades para implementar. Aquí se realiza el cierre del proyecto, donde se prepara la liberación del videojuego, se verifican las versiones a entregar, se genera la documentación final y se realiza el pre-lanzamiento y el lanzamiento. Al contrario XGD cuenta con más fases, haciendo complejo el desarrollo del videojuego y aún más si son equipos sin experiencia en la utilización de la metodología, estas fases son: Exploración, Planificación, Iteración, Producción, Mantenimiento y Muerte del Proyecto.</p>
Actividades	<p>Tanto la metodología Scrum como la metodología DAV plantean actividades muy prácticas, fuertemente ligadas, ayudando a desarrollar de manera rápida, fácil y con una amplia visión de lo que se está haciendo, todas sus actividades son importantes, siendo estas: Elaborar las funcionalidades del videojuego que se encuentra previstas en una iteración, Integrar las funcionalidades desarrolladas en el videojuego, Revisar las funcionalidades desarrolladas en la iteración para corregir posibles errores y Ajustar encajándolas correctamente; estas actividades no tienen una secuencia debido a que a veces un elemento de la Reserva del Producto se tiene que desarrollar, integrar, y revisar cuando otro sólo debe ser revisado o ajustado. en cambio XGD parte con una pequeña visión al inicio fortaleciendo y complementando el desarrollo del videojuego según este vaya avanzando y siguiendo el orden establecido de las actividades planteadas tales como: Escuchar a los clientes las funcionalidades que quieren que contenga el videojuego, Diseñar una buena estructura que</p>

	<p>organice al desarrollo del videojuego permitido que este crezca de manera sencilla, Codificar plasmando las ideas y funcionalidades del videojuego planteadas por el cliente a través del código y hacer Pruebas para saber si lo implementado es lo deseado; siendo la más importante la de Codificar.</p>
Artefactos	<p>La metodología SCRUM propone cinco artefactos: User Story, Product Backlog, Sprint Backlog, Burn-Down y Burn-Up, para ayudar a planificar y revisar cada uno de los Sprints, aportando medios ineludibles para efectuar cada una de las reuniones, además permiten saber en cualquier momento qué tenemos que hacer, qué estamos haciendo y como estamos avanzando en base al tiempo estimado, de este modo se tiene un control integral del proyecto, en el caso de la metodología XGD propone cuatro artefactos: Tablero XGD, User Story (Prueba de Aceptación del User Story), Task Card, Card CRC, de esta manera da a conocer a detalle los user stories proporcionando al equipo de desarrollo una mejor visión de lo que se quiere hacer a través de la descripción de las tareas, permitiendo al equipo completo contribuir en la tarea del diseño del videojuego y planificar las iteraciones, pero no brinda artefactos que lleven un monitoreo de lo que se está realizando en base a los tiempos establecidos. La metodología DAV al ser una metodología adaptada se complementa fortaleciéndose de los artefactos tanto de SCRUM como de XGD, siendo estos: Historia de Usuario, Prueba de Aceptación, Reserva del Producto, Tarjetas de Tarea, Reserva de Iteración, Burn-Down, Burn-Up y las Cartas DAV; de esta manera posee una mayor eficiencia, facilitando la gestión y desarrollo del proyecto.</p>
	<p>La metodología SCRUM utiliza herramientas excelentes con distribuciones libres o pagadas para facilitar la gestión y seguimiento del proyecto ayudando a manejar y evitar problemas que surjan en el transcurso del</p>

Herramientas	desarrollo o liberación del videojuego, permitiendo gestionar el Product Backlog, Sprint Backlog y los User Stories, además permiten un seguimiento de la velocidad y avance de los Sprints. Del mismo modo la metodología XGD utiliza herramientas para la gestión, integración, medición y refactorización de código. La metodología DAV utiliza las herramientas tanto de SCRUM como de XGD, facilitando la gestión y el desarrollo del proyecto.
---------------------	---

Por todo lo expuesto anteriormente se concluye que la adaptación de la nueva metodología en el desarrollo de videojuegos facilita la creación de juegos en la plataforma Android de forma técnica y sencilla, comprobando y aceptando así la hipótesis planteada.

CONCLUSIONES

- La investigación de las metodologías Ágiles Scrum y XGD permite conocer el proceso, fases, artefactos, prácticas, actividades, roles, reuniones y herramientas que poseen cada una de ellas, enfocadas al desarrollo de videojuegos.
- La investigación teórica de las metodologías ágiles Scrum, XGD y de los prototipos desarrollados con cada metodología, permite llevar a cabo la adaptación de una nueva metodología para desarrollo de videojuegos.
- En base a las características, encontramos que la metodología SCRUM se enfoca en la gestión del proyecto aportando a las organizaciones el mayor valor posible a corto plazo, con resultados de calidad que responden a las necesidades reales del negocio, no posee prácticas de ingeniería a diferencia de XGD que es una metodología de ingeniería que contiene prácticas que aseguran la calidad focalizándose en el código, la simplicidad, la comunicación, el valor y la retroalimentación.
- En base a todos los parámetros y criterios generales se ha comprobado que las dos metodologías se basan en el desarrollo iterativo e incremental, pero Scrum se enfoca en la gestión del proyecto, no posee prácticas de ingeniería, alcanzando un porcentaje final de 80,77%, a diferencia de XGD que es una metodología de ingeniería que contiene prácticas que aseguran la calidad focalizándose en el código, alcanzando un porcentaje final de 86,53%, calificando a las dos metodologías como Muy Buenas.
- La adaptación de las metodologías Ágiles Scrum y Extreme Game Development en la nueva metodología para desarrollo de videojuegos, incrementa la productividad de código desarrollado, la gestión del proyecto; logrando llegar en tiempo y forma a los plazos requeridos con versiones jugables del videojuego, permitiendo tener un mayor control del avance del proyecto, tanto al cliente

como a los desarrolladores para reducir la probabilidad de cambios de requerimientos en forma tardía.

- La nueva metodología adaptada a diferencia de Scrum y XGD en la estimación de las Historias de Usuario limita como máximo una duración de 16 horas al utilizar las cartas DAV, de esta manera una tarea no supera los dos días de trabajo facilitando su desarrollo.
- La utilización de la metodología adaptada en la creación del videojuego "El tren del conocimiento" en Android, facilita el desarrollo de forma técnica y sencilla.

RECOMENDACIONES

- Se sugiere a los desarrolladores de videojuegos estudiar la metodología adaptada para conocer todo el proceso que posee y su terminología, antes de ser puesta en práctica.
- Estimar y priorizar correctamente las Historias de Usuario para evitar problemas y retrasos en el desarrollo del videojuego.
- Los desarrolladores de videojuegos deben utilizar la nueva metodología adaptada DAV siguiendo el proceso en su totalidad, debido a que facilita la creación de videojuegos de forma técnica y sencilla, obteniendo resultados eficientes.
- La colaboración del usuario final y/o cliente en todo el desarrollo del videojuego es indispensable para poder aplicar la metodología.
- Los miembros del equipo de desarrollo deben ser expertos y avanzados en su disciplina, con vocación para trabajar en equipo.

RESUMEN

Adaptación de las metodologías de desarrollo ágil Scrum y XGD (Extreme Game Development) en una metodología para desarrollo de videojuegos en la plataforma Android. Caso práctico: desarrollo de un videojuego para la Escuela Mixta Dr. Guido Mera Vera.

Se aplicó el método analítico e investigativo en el análisis de cada metodología para obtener los resultados de los parámetros de cada criterio de acuerdo a las pruebas realizadas en el desarrollo de un prototipo de videojuego con cada una de las metodologías. Para la demostración de la hipótesis no se utilizó ningún método estadístico debido a que no se posee una población.

Las herramientas utilizadas para el desarrollo y diseño fueron las siguientes: Entorno de Android en la versión 2.3.3, editor online de imágenes gratuito Pixlr y generador de textos y logos online gratuito Cool Text, usadas en las portátiles.

De acuerdo a los resultados obtenidos en base al análisis realizado a las dos metodologías previo a la adaptación de la nueva metodología se obtuvo el siguiente resultado: en el criterio “Planificación” Scrum obtuvo el 91.77% y XGD el 75%, en el criterio “Ciclo de Vida del Proyecto” Scrum obtuvo el 93.75% y XGD el 75%, en el criterio “Calidad” Scrum obtuvo el 66.67% y XGD el 100%, y en el criterio “Herramientas Utilizadas en el Desarrollo” Scrum obtuvo el 66.67% y XGD el 100%.

Entonces se concluye que Scrum se enfoca en la gestión del proyecto, no posee prácticas de ingeniería a diferencia de XGD que es una metodología de ingeniería que contiene prácticas que aseguran la calidad focalizándose en el código.

Por lo tanto combinar Scrum con Extreme Game Development en la nueva metodología incrementa la calidad y productividad de código desarrollado, la gestión del proyecto con un seguimiento estricto en la planificación, tareas, iteraciones y tiempos; planteando fases y actividades muy prácticas, fuertemente ligadas, ayudando a desarrollar de manera rápida con una amplia visión de lo que se está haciendo.

Se recomienda a los desarrolladores de videojuegos que al utilizar la nueva metodología adaptada se siga el proceso en su totalidad debido a que facilita la creación de videojuegos de forma técnica y sencilla.

SUMMARY

The purpose of this investigation is to adjust Scrum and XGD methodologies of agile development (Extreme Game Development), to an Android video-game platform for a methodology development. A practical case: A video-game development at Escuela Mixta Dr. Guido Mera Vera.

Two methodologies, i.e. analytic and research were applied in the analysis of each methodology for obtaining parameter results concerning test criteria which were carried out on a video-game prototype development. No statistical methodology for hypothesis proof was used because there was a lack of population.

For developing and designing, the following tools were used: version 2.3.3. Android environment, free Pixlr image online editor, free Cool Text online logo text generator, which were used on the laptops.

According to obtained results based on analysis performed on two methodologies, previous to the new methodology adjustment, the following was reached: Scrum "Planning" criterion 91.77%, XGD criterion 75%, Scrum "Project Live Cycle" criterion 93.75%, XGD criterion 75%, Scrum "Quality" 66.67%, XGD 100%, Scrum "Tools Utilized in the Development" 66.67%, and XGD 100%.

It can be concluded that Scrum focuses on project management which does not possess engineering practice like the XGD method does, the XGD is an engineering methodology which contains practices that ensure focalized code quality.

Therefore, combining Scrum with Extreme Game Development to the new methodology increases the following development code quality and productivity, project management with strict planning tracing, tasks, time and interaction; setting phases and practical activities strongly linked, helping this way to quickly develop such methodology with ample vision of what is been doing.

It is recommended that video-game developers who use a new DAV (Desarrollo Ágil de Video-juegos) methodology follow the whole process because it makes video creation easy to follow in a more technical and simpler way.

GLOSARIO

ACID: Conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

Actividades: Son tareas que deben ejecutarse para llegar en conjunto a un objetivo preestablecido o deseado.

ADB: (Android Debug Bridge). Es una herramienta que viene junto con el SDK de Android y nos permite acceder y controlar un dispositivo Android desde una PC.

ADT: (Android Developer Tools). Es un plugin para el IDE Eclipse, proporciona un entorno de desarrollo de nivel profesional para la creación de aplicaciones de Android. Es un IDE Java completo con funciones avanzadas para ayudar a construir, probar, depurar y empaquetar las aplicaciones de Android.

API: (Interfaz de Programación de Aplicaciones). Conjunto de funciones y procedimientos (o métodos en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

APK: (Android Application Package File). Es el formato de archivo utilizado para distribuir e instalar software en el Sistema Operativo Android.

Arquitectura: Es la estructura de un sistema que incluye componentes de software, las propiedades visibles externas de esos componentes y las relaciones entre estos.

Bytecode: Es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un archivo binario que contiene un programa ejecutable similar a un módulo objeto, que es un archivo binario producido por el compilador cuyo contenido es el código objeto o código máquina.

Ciclo de Vida de una Metodología: Describe el desarrollo en un proyecto o software, desde la fase inicial hasta la fase final.

Código Libre: Es el término con el que se conoce al software distribuido y desarrollado libremente

Compilación Just-in-Time: Es una técnica para mejorar el rendimiento de sistemas de programación que compilan a bytecode, consistente en traducir el bytecode a código máquina nativo en tiempo de ejecución.

Diseño: Se refiere a un boceto, bosquejo o esquema que se realiza en un soporte material, antes de concretar la producción del videojuego.

Estimación: Predicción basada en un modelo probabilístico, es decir, la cantidad que se esté estimando puede tomar no solamente un valor sino distintos valores, además están asociadas con el esfuerzo, costo y el tiempo de las actividades identificadas del proyecto.

Etapas: Cada una de las partes de las que se compone el ciclo de vida de una metodología.

Figuras Bidimensionales Planas: Tienen como características dos dimensiones: ancho y alto, es decir no tienen volumen.

Framework: Conjunto integrado de componentes que colaboran para proporcionar una arquitectura reutilizable para una familia de aplicaciones.

Gestión de Proyectos: Es el proceso por el cual se planifica, dirige y controla el desarrollo de un sistema aceptable con un costo mínimo y dentro de un período de tiempo específico

Google Play: Es una tienda de software en línea desarrollada por Google para los dispositivos con sistema operativo Android.

GUI: (Interfaz Gráfica de Usuario). Conjunto de formas y métodos que posibilitan la interacción de un sistema con los usuarios utilizando formas gráficas e imágenes.

Interfaz: Conjunto de servicios proporcionados por una clase.

Marco de Trabajo Ágil: Se basan en principios ágiles que proponen una forma alternativa de administrar proyectos altamente complejos.

Mantenibilidad: Cualidad de una aplicación que hace que el mantenimiento sea fácil.

OpenGL: Es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.

Principios: Son reglas o normas que orientan en el desarrollo de un proyecto.

Procesos: Son actividades que se caracterizan por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema.

Programación Orientada a Objetos: Es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos.

Prototipo: Versión reducida del videojuego final. Se construye para obtener requisitos o incrementos hasta su versión final.

Roles: Son funciones encargadas a cada miembro del equipo en el desarrollo de un proyecto.

Recursos: Son los elementos utilizados para poder realizar la ejecución de cada una de las tareas.

Requerimiento: Es una necesidad documentada sobre el contenido, forma o funcionalidad de un videojuego y muestran qué elementos y funciones son necesarias.

Retorno Sobre la Inversión: Es un estimado del beneficio (retorno) sobre el dinero gastado (inversión) en una alternativa en particular, y consiste en determinar los beneficios, calcular los costes y resumir los resultados.

Reunión: Acto en que se reúnen todos o parte de los miembros el equipo de desarrollo particularmente para tratar algún asunto.

Riesgo: Un evento o condición incierta que, si se produce, tiene un efecto positivo o negativo en los objetivos de un proyecto.

SDK: (Software Development Kit). Es un kit de desarrollo de software con el cual podremos desarrollar aplicaciones y ejecutar un emulador del sistema Android de la versión que sea.

SQLite: Es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña

Time box: Fija el tiempo máximo para conseguir los objetivos planteados al inicio del desarrollo de un proyecto y tomar decisiones según su avance.

TIC: Son un conjunto de servicios, redes, software, aparatos que tienen como fin el mejoramiento de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.

Valores: Son convicciones profundas que permiten orientar el comportamiento de las personas en el desarrollo de un proyecto.

Versión: Una parte del videojuego final en una determinada Iteración o Sprint.

Versiones jugables: Son entregas jugables o funcionales de un videojuego en lapsos de tiempos pequeños

BIBLIOGRAFÍA

1. **BECK, K.**, Extreme Programming Explained: Embrace Change., Canada., Addison-Wesley., 2000., pp 121-164.
2. **BECK, K. y FOWLER, M.**, Planning Extreme Programming., Canada., Addison-Wesley., 2001., pp 63-124.
3. **KEITH, C.**, Agile Game Development with Scrum., Indiana – United States of America., Addison-Wesley., 2010., 367 p.

BIBLIOGRAFÍA DE INTERNET

4. **CARACTERÍSTICAS Y APLICACIONES DE ANDROID**
<http://www.configurarequipos.com/doc1107.html>
2009/04/02
5. **CICLO DE VIDA DE UN PROYECTO XP**
<http://oness.sourceforge.net/proyecto/html/ch05s02.html>
2004/09/28
6. **DESARROLLO DE APLICACIONES PARA
DISPOSITIVOS CON SISTEMA OPERATIVO ANDROID**
<http://riunet.upv.es/bitstream/handle/10251/10299/Memoria.pdf>
2011/01/21

**7. DESARROLLO DE VIDEOJUEGOS SOBRE LA
PLATAFORMA ANDROID**

[http://upcommons.upc.edu/pfc/bitstream/2099.1/14016/1/
76880.pdf](http://upcommons.upc.edu/pfc/bitstream/2099.1/14016/1/76880.pdf)

2012/01/10

8. DISEÑO DE VIDEOJUEGOS JAVA 2D

<http://www.xcubo.es/2drazing/doc/Java2D.pdf>

2010

9. EL MODELO SCRUM

http://www.navegapolis.net/files/s/NST-010_01.pdf

2006

**10. EXTREME GAME DEVELOPMENT: JUSTO A TIEMPO,
CADA VEZ**

[http://www.gamasutra.com/view/feature/2827/extreme_ga
me_development_right_on_.php](http://www.gamasutra.com/view/feature/2827/extreme_game_development_right_on_.php)

2003/07/16

11. EXTREME PROGRAMMING FOR GAMES

<http://c2.com/cgi/wiki?ExtremeProgrammingForGames>

2007-06-13

**12. EXTREME PROGRAMMING IDEAL PARA EL
DESARROLLO DEL JUEGO**

[http://www.gamasutra.com/view/feature/1702/embracing
fun_why_extreme_.php](http://www.gamasutra.com/view/feature/1702/embracing_fun_why_extreme_.php)

2007/03/01

13. EXTREME PROGRAMMING IN GAME DEVELOPMENT

<http://prezi.com/blcyk2mdg-as/extreme-programming-in-game-development/>

2011/12/08

14. FLEXIBILIDAD CON SCRUM

http://www.scrummanager.net/files/flexibilidad_con_scrum_ebook.pdf

2008/10/01

15. GAME-SCRUM: AN APPROACH TO AGILE GAME DEVELOPMENT

http://sbgames.org/sbgames2010/proceedings/computing/sport/Computing_short19.pdf

2010

16. HISTORIA DE LOS VIDEOJUEGOS

http://www.esdelibro.es/archivos/trabajos10/201000323_videojuegos_trabajo.pdf

2008-07-01

17. METODOLOGÍAS ÁGILES: SCRUM Y TÉCNICAS DE ESTIMACIÓN ÁGIL

<http://jorgemanrubia.net/blog/wp-content/uploads/2009/06/2009-06-CharlaPreparaticAgil.pdf>

2009/06/01

18. PROGRAMACIÓN ANDROID

http://www.sgoliver.net/blog/?page_id=3011

2013

19. PROGRAMACIÓN EN ANDROID PARA PRINCIPIANTES

<https://dl.dropboxusercontent.com/u/1469503/Curso%20de%20programaci%C3%B3n%20b%C3%A1sico%20de%20Android%20-%20FAQSAndroid.pdf>

2012/08/22

20. SCRUM Y XP DESDE LAS TRINCHERAS

<http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf>

2007

21. XGD: EXTREME PROGRAMMING FOR GAME DEVELOPERS

<http://www.onlamp.com/pub/wlg/3668>

2003/08/13

ANEXOS

ANEXO I

USER STORIES, CASOS DE PRUEBA DE ACEPTACIÓN, TASK CARD Y CRC DEL PROTOTIPO “XGD”.

User Story Primera Iteración

USER STORY	
Numero: US06	Nombre: Esquema de bloques
Usuario: Diseñador	
Numero de modificación del story: N/A	Iteración Asignada: Primera
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Diseñador, quiero que los bloques formen un esquema, de modo que se encuentren separados para que exista dificultad en el juego.	
Observaciones:	

USER STORY	
Numero: US07	Nombre: Movimiento de bola
Usuario: Jugador	
Numero de modificación del story: N/A	Iteración Asignada: Primera
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Jugador quiero que la bola se mueva por toda la pantalla rebotando en cada lado para que pueda romper los bloques.	
Observaciones:	

USER STORY	
Numero: US10	Nombre: Mover Pala
Usuario: Jugador	
Numero de modificación del story: N/A	Iteración Asignada: Primera
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Jugador quiero que la pala se encuentre en la parte inferior, y pueda moverse en forma horizontal de modo que la bola no se caiga y revolarla contra los bloques para romperlos.	
Observaciones:	

USER STORY	
Numero: US11	Nombre: Numero de Vidas
Usuario: Diseñador	
Numero de modificación del story: N/A	Iteración Asignada: Primera
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Diseñador quiero que el jugador posea solo 3 vidas de modo que al agotarse las vidas se termine el juego y muestre al final un mensaje que indique que perdió el juego.	
Observaciones:	

USER STORY 13	
Numero: US13	Nombre: Finalizar juego
Usuario: Diseñador	
Numero de modificación del story: N/A	Iteración Asignada: Primera
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Diseñador quiero que al romper todos los bloques finalice el juego para indicar que ha terminado el juego y ha ganado.	
Observaciones:	

User Story Segunda Iteración

USER STORY	
Numero: US01	Nombre: Instrucciones
Usuario: Jugador	
Numero de modificación del story: N/A	Iteración Asignada: Segunda
Prioridad en Negocio: Media	Riesgos en desarrollo: Baja
Descripción: Como jugador quiero visualizar las instrucciones del juego para saber cómo jugar.	
Observaciones:	

USER STORY	
Numero: US02	Nombre: Créditos
Usuario: Jugador	
Numero de modificación del story: N/A	Iteración Asignada: Segunda
Prioridad en Negocio: Media	Riesgos en desarrollo: Baja
Descripción: Como Jugador, quiero visualizar los créditos del juego, para saber quién lo desarrolló.	
Observaciones:	

USER STORY	
Numero: US03	Nombre: Menú principal
Usuario: Jugador	
Numero de modificación del story: N/A	Iteración Asignada: Segunda
Prioridad en Negocio: Alta	Riesgos en desarrollo: Media
Descripción: Como jugador quiero tener un menú principal para que pueda ingresar a jugar, visualizar las instrucciones, mostrar los créditos del videojuego y salir.	
Observaciones:	

USER STORY	
Numero: US04	Nombre: Botones Regresar
Usuario: Diseñador	
Numero de modificación del story: N/A	Iteración Asignada: Segunda
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Diseñador, quiero que exista en la pantalla de créditos como en la de instrucciones un botón de regreso, de modo que pueda volver al menú principal	
Observaciones:	

USER STORY	
Numero: US05	Nombre: Aspecto de Bola
Usuario: Diseñador	
Numero de modificación del story: N/A	Iteración Asignada: Segunda
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Diseñador, quiero que la bola tenga el aspecto de una bola de jugar futbol, para que se muestre más divertida.	
Observaciones:	

USER STORY	
Numero: US08	Nombre: Movimiento de bola
Usuario: Jugador	
Numero de modificación del story: N/A	Iteración Asignada: Primera
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Jugador, quiero que al romper los bloques con la bola, se escuche un sonido de "golpe o rotura", para que sea divertido y le de realismo.	
Observaciones:	

USER STORY	
Numero: US09	Nombre: Puntaje
Usuario: Jugador	
Numero de modificación del story: N/A	Iteración Asignada: Segunda
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como jugador quiero visualizar el puntaje de los bloques que voy rompiendo de modo que pueda conocer cuántos puntos voy acumulando.	
Observaciones:	

USER STORY	
Numero: US12	Nombre: Numero de Intentos
Usuario: Jugador	
Numero de modificación del story: N/A	Iteración Asignada: Segunda
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Jugador quiero ver mediante un gráfico las vidas para conocer los intentos que tengo.	
Observaciones:	

USER STORY	
Numero: US14	Nombre: Exista Audio en el menú.
Usuario: Diseñador	
Numero de modificación del story: N/A	Iteración Asignada: Segunda
Prioridad en Negocio: Alta	Riesgos en desarrollo: Baja
Descripción: Como Diseñador quiero que exista audio de fondo en la pantalla de menú como en el juego para que sea divertido.	
Observaciones:	

Casos de Prueba de Aceptación primera Iteración

Caso de Prueba de Aceptación	
Código: CP06	Historia de Usuario(Nro. y Nombre): US06- Diseñador
Nombre: Aspecto de Bola	
Descripción: Como Diseñador, quiero que los bloques formen un esquema, de modo que se encuentren separados para que exista dificultad en el juego.	
Condiciones de Ejecución: Los bloques deben formar un esquema con el propósito de que exista dificultad.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Juego. Los bloques aparecerán formando un esquema.	
Resultado Esperado: Funcionalidad correcta en la ejecución de todos los pasos.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de Prueba de Aceptación	
Código: CP07	Historia de Usuario(Nro. y Nombre): US07- Jugador
Nombre: Movimiento de bola	
Descripción: Como Jugador, quiero que la bola se mueva por toda la pantalla, rebotando en cada lado para que pueda romper los bloques.	
Condiciones de Ejecución:	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Juego. Pulsar sobre la bola para empezar a romper los bloques. La bola rompe los bloques rebota sobre las paredes de la pantalla. La bola toma trayectorias diferentes.	
Resultado Esperado: Funcionalidad correcta en la ejecución de todos los pasos.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP10	Historia de Usuario(Nro. y Nombre): US10- Jugador
Nombre: Mover Pala	
Descripción: Como Jugador quiero que la pala se encuentre en la parte inferior, y pueda moverse en forma horizontal de modo que la bola no se caiga y revolarla contra los bloques para romperlos.	
Condiciones de Ejecución: Pulsar sobre la pala y moverla de forma horizontal con el fin de evitar que la bola caiga.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Videojuego. Pulsar sobre la pala.	
Resultado Esperado: Funcionalidad correcta en el movimiento de la pala y buscando que rebote la pelota al haber contacto.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP11	Historia de Usuario(Nro. y Nombre): US11- Diseñador
Nombre: Numero de Vidas	
Descripción: Como Diseñador quiero que el jugador posea solo 3 vidas de modo que al agotarse las vidas se termine el juego y muestre al final un mensaje que indique que perdió el juego.	
Condiciones de Ejecución: Cuando la bola caiga o la pala no pueda devolverla se perderá una vida descontándola de las iniciales, cuando ya no tenga vidas el juego finaliza y aparece un mensaje.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Videojuego. Pulsar sobre la pala. La bola sale desprendida. La pala no puede devolver a la bola. La bola cae 3 veces. El juego finaliza. Aparece un mensaje en la pantalla.	
Resultado Esperado: Funcionalidad correcta en el movimiento de la pala y buscando que rebote la pelota al haber contacto.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP13	Historia de Usuario(Nro. y Nombre): US13- Diseñador
Nombre: Numero de Intentos	
Descripción: Como Diseñador quiero que al romper todos los bloques finalice el juego para indicar que ha terminado el juego y ha ganado.	
Condiciones de Ejecución: Cuando la bola rompa todos los bloques y no hayas perdido todas tus vidas el juego finaliza y muestra un mensaje que te afirma que has ganado.	
Entrada / Pasos de Ejecución: Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Videojuego. Pulsar sobre la pala. La bola sale despedida. La bola choca contra un bloque este desaparece. La bola toma una trayectoria diferente. La bola sigue rompiendo bloques. La bola rompe el último bloque. El jugador tiene vidas. El juego finaliza y aparece un mensaje en donde le dice al jugador que ha ganado.	
Resultado Esperado: Funcionalidad correcta en el movimiento de la pala y buscando que rebote la pelota al haber contacto.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Casos de Prueba de Aceptación primera Iteración

Caso de prueba de Aceptación	
Código: CP01	Historia de Usuario(Nro. y Nombre): US01- Jugador
Nombre: Instrucciones	
Descripción: Como jugador quiero visualizar las instrucciones del juego para saber cómo jugar.	
Condiciones de Ejecución: Pulsar sobre el botón del Menú Instrucciones para conocer el correcto funcionamiento del juego.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Instrucciones. Leer las instrucciones. Pulsar sobre el botón regresar.	
Resultado Esperado: Funcionalidad correcta en la ejecución de todos los pasos.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP02	Historia de Usuario(Nro. y Nombre): US02- Jugador
Nombre: Créditos	
Descripción: Como jugador quiero visualizar los créditos del juego para saber quién lo desarrolló.	
Condiciones de Ejecución: Pulsar sobre el botón del Menú Créditos y aparecerá una pantalla con la descripción de quien desarrollo el juego.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Créditos. Leer los créditos.	
Resultado Esperado: Funcionalidad correcta en la ejecución de todos los pasos.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP03	Historia de Usuario(Nro. y Nombre): US03- Jugador
Nombre: Menú principal	
Descripción: Como jugador quiero tener un menú principal para Pueda ingresar a jugar, visualizar las instrucciones, mostrar los créditos del videojuego y salir.	
Condiciones de Ejecución: Pulsar sobre los botones del menú según las necesidades que el jugador disponga.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre los botones que aparecen.	
Resultado Esperado: Funcionalidad correcta en la ejecución de todos los pasos.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP04	Historia de Usuario(Nro. y Nombre): US04- Diseñador
Nombre: Botones Regresar	
Descripción: Como Diseñador, quiero que exista en la pantalla de créditos como en la de instrucciones un botón de regreso, de modo que pueda volver al menú principal.	
Condiciones de Ejecución: Pulsar sobre el botón del menú Instrucciones aparecerá la pantalla de instrucciones pulsar sobre el botón regresar para ir al menú principal, de igual forma pulsar sobre el botón del menú créditos aparecerá la pantalla de créditos, pulsar sobre el botón regresar para ir al menú principal.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Créditos o Instrucciones. Leer los créditos o Instrucciones. Pulsar sobre el botón regresar de la pantalla créditos o instrucciones para ir al menú principal.	
Resultado Esperado: Funcionalidad correcta en la ejecución de todos los pasos.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP05	Historia de Usuario(Nro. y Nombre): US05- Diseñador
Nombre: Aspecto de Bola	
Descripción: Como Diseñador, quiero que la bola tenga el aspecto de una bola de jugar futbol, para que se muestre más divertida.	
Condiciones de Ejecución: El aspecto de la bola debe parecerse a un balón de jugar futbol.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Juego. Aparecerá la bola con aspecto de balón de jugar futbol.	
Resultado Esperado: Funcionalidad correcta en la ejecución de todos los pasos.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP08	Historia de Usuario(Nro. y Nombre): 08- Jugador
Nombre: Romper Bloques	
Descripción: Como Jugador, quiero que al romper los bloques con la bola, se escuche un sonido de "golpe o rotura", para que sea divertido y le de realismo.	
Condiciones de Ejecución: Al momento de impactarla bola con el bloque se escucha un sonido, el bloque desaparece y la bola toma otra trayectoria.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Videojuego. Pulsar sobre la pala. La bola sale despedida. La bola choca contra un bloque este desaparece. La bola toma una trayectoria diferente.	
Resultado Esperado: Funcionalidad correcta en el impacto de la bola con el bloque y sonido exacto.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP09	Historia de Usuario(Nro. y Nombre): US09- Jugador
Nombre: Puntaje	
Descripción: Como jugador quiero visualizar el puntaje de los bloques que voy rompiendo de modo que pueda conocer cuántos puntos voy acumulando.	
Condiciones de Ejecución: Cuando la bola rompe un bloque el puntaje aumenta proporcionalmente.	
Entrada / Pasos de Ejecución Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Videojuego. Pulsar sobre la pala. La bola sale despedida. La bola choca contra un bloque este desaparece. La puntuación aumenta proporcionalmente.	
Resultado Esperado: Funcionalidad correcta cuando al romper el bloque con la bola el puntaje aumenta proporcionalmente.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP12	Historia de Usuario(Nro. y Nombre): US12- Jugador
Nombre: Numero de Intentos	
Descripción: Como Jugador quiero ver mediante un gráfico las vidas para conocer los intentos que tengo.	
Condiciones de Ejecución: Cuando la bola caiga los gráficos que representan a las vidas van desapareciendo una a una en caso de quedar sin ellas el jugador pierde automáticamente.	
Entrada / Pasos de Ejecución: Ingresar al juego. Aparece el menú de inicio. Pulsar sobre el Botón Iniciar Videojuego. Pulsar sobre la pala. La bola sale despedida. La bola choca contra un bloque este desaparece. La bola toma una trayectoria diferente. La bola cae al no ser golpeada con la pala. Las vidas se eliminan una a una.	
Resultado Esperado: Funcionalidad correcta en el movimiento de la pala y buscando que rebote la pelota al haber contacto.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Caso de prueba de Aceptación	
Código: CP14	Historia de Usuario(Nro. y Nombre): US14- Diseñador
Nombre: Exista Audio en el menú.	
Descripción: Como Diseñador quiero que exista audio de fondo en la pantalla de menú como en el juego para que sea divertido.	
Condiciones de Ejecución: Cuando aparece la pantalla del menú del videojuego se escucha un sonido de fondo con el fin de hacer más agradable el juego.	
Entrada / Pasos de Ejecución: Ingresar al juego. Aparece el menú de inicio. Se escucha un sonido de fondo.	
Resultado Esperado: Funcionalidad correcta en el movimiento de la pala y buscando que rebote la pelota al haber contacto.	
Evaluación de la Prueba: Debe estar realizado todos los pasos detallados anteriormente.	

Task Card Primera Iteración

Tarea de Ingeniería	
Número de Tarea: T06-01	Historias de Usuario (Nro. y Nombre): US06- Diseñador
Nombre Tarea: Imagen Bloque	
Tipo de Tarea: Diseño	Puntos Estimados: 3
Fecha Inicio: 21 de Enero de 2013	Fecha Fin: 21 de Enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: El bloque tiene un aspecto de edificio con el fin de dar mejor aspecto al juego	

Tarea de Ingeniería	
Número de Tarea: T06-02	Historias de Usuario (Nro. y Nombre): US06- Diseñador
Nombre Tarea: Esquema Bloques	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 21 de Enero de 2013	Fecha Fin: 21 de Enero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: Los bloques formaran un esquema para dar un poco de dificultad al juego.	

Tarea de Ingeniería	
Número de Tarea: T07-01	Historias de Usuario (Nro. y Nombre): US07- Jugador
Nombre Tarea: Movimiento	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 22 de Enero de 2013	Fecha Fin: 22 de Enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: La bola sale desprendida de la pala para iniciar el juego moviéndose en toda la pantalla hasta encontrar algún objeto para cambiar de trayectoria.	

Tarea de Ingeniería	
Número de Tarea: T07-02	Historias de Usuario (Nro. y Nombre): US07- Jugador
Nombre Tarea: Rebotar	
Tipo de Tarea: Diseño	Puntos Estimados: 4
Fecha Inicio: 22 de Enero de 2013	Fecha Fin: 22 de Enero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: Al chocar con las paredes de la pantalla la bola rebota cambiando su trayectoria al igual que al chocar con los bloques.	

Tarea de Ingeniería	
Número de Tarea: T10-01	Historias de Usuario (Nro. y Nombre): US10- Jugador
Nombre Tarea: Mover Pala	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 23 de Enero de 2013	Fecha Fin: 23 de Enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: Mover la pala de izquierda a derecha pulsando sobre la misma con el fin de ganar el mayor puntaje posible.	

Tarea de Ingeniería	
Número de Tarea: T11-01	Historias de Usuario (Nro. y Nombre): US11- Diseñador
Nombre Tarea: Perder Vidas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 24 de Enero de 2013	Fecha Fin: 24 de Enero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: Pulsar sobre la pala para iniciar a jugar, la bola sale desprendida, la bola cae y la pala no puede devolver el golpe, esto pasa tres veces. El juego finaliza y aparece un mensaje.	

Tarea de Ingeniería	
Número de Tarea: T13-01	Historias de Usuario (Nro. y Nombre): US13- Diseñador
Nombre Tarea: Ganar	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 25 de Enero de 2013	Fecha Fin: 25 de Enero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: cuando la bola rompa todos los bloques aparece un mensaje que el jugador ha ganado.	

Tarea de Ingeniería	
Número de Tarea: T13-01	Historias de Usuario (Nro. y Nombre): US13- Diseñador
Nombre Tarea: Finalizar juego	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 25 de Enero de 2013	Fecha Fin: 25 de Enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: Cuando la bola rompa todos los bloques aparece un mensaje que el jugador ha ganado y finaliza el juego.	

Task Card Segunda Iteración

Tarea de Ingeniería	
Número de Tarea: T01-01	Historias de Usuario (Nro. y Nombre): US01- Jugador
Nombre Tarea: Pantalla Instrucciones	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 28 de Enero de 2013	Fecha Fin: 28 de Enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: La pantalla de instrucciones ayuda a conocer el desarrollo del juego para tener una idea concreta del mismo.	

Tarea de Ingeniería	
Número de Tarea: T02-01	Historias de Usuario (Nro. y Nombre): US01- Jugador
Nombre Tarea: Pantalla Créditos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 28 de Enero de 2013	Fecha Fin: 28 de Enero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: La pantalla de créditos ayuda a conocer a quien o quienes desarrollaron el juego.	

Tarea de Ingeniería	
Número de Tarea: T03 -01	Historias de Usuario (Nro. y Nombre): US03- Jugador
Nombre Tarea: Botones	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 29 de Enero de 2013	Fecha Fin: 29 de Enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: Los botones son apropiados en modelo, tamaño estructura cumpliendo funcionalidad.	

Tarea de Ingeniería	
Número de Tarea: T03 -02	Historias de Usuario (Nro. y Nombre): US03- Jugador
Nombre Tarea: Fondo Menú	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 29 de Enero de 2013	Fecha Fin: 29 de Enero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: El fondo de la pantalla del menú principal está diseñado de acuerdo a la intuición del programador, diseñador.	

Tarea de Ingeniería	
Número de Tarea: T04-01	Historias de Usuario (Nro. y Nombre): US04- Diseñador
Nombre Tarea: Salir pantalla Créditos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 30 de Enero de 2013	Fecha Fin: 30 de Enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: El botón regresar de la pantalla de créditos sirve para salir e ir al menú principal.	

Tarea de Ingeniería	
Número de Tarea: T04-02	Historias de Usuario (Nro. y Nombre): US04- Diseñador
Nombre Tarea: Salir pantalla Instrucciones	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 30 de Enero de 2013	Fecha Fin: 30 de Enero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: El botón regresar de la pantalla de instrucciones sirve para salir e ir al menú principal.	

Tarea de Ingeniería	
Número de Tarea: T05-01	Historias de Usuario (Nro. y Nombre): US05- Diseñador
Nombre Tarea: Diseño Bola	
Tipo de Tarea: Diseño	Puntos Estimados: 3
Fecha Inicio: 31 de Enero de 2013	Fecha Fin: 31 de Enero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: La bola que aparece en el juego es un balón de jugar futbol.	

Tarea de Ingeniería	
Número de Tarea: T08-01	Historias de Usuario (Nro. y Nombre): US08- Jugador
Nombre Tarea: Desaparecer Bloque	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 01 de Febrero de 2013	Fecha Fin: 01 de Febrero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: Pulsar sobre la pala para desprender la bola esta choca contra el bloque el mismo que desaparece.	

Tarea de Ingeniería	
Número de Tarea: T08-02	Historias de Usuario (Nro. y Nombre): US08- Jugador
Nombre Tarea: Sonido de impacto	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 01 de Febrero de 2013	Fecha Fin: 01 de Febrero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: Cuando la choca la bola con el bloque se escucha un sonido.	

Tarea de Ingeniería	
Número de Tarea: T09 -01	Historias de Usuario (Nro. y Nombre): US09- Jugador
Nombre Tarea: Puntaje	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 01 de Febrero de 2013	Fecha Fin: 01 de Febrero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: Cuando el jugador no puede devolver la bola y esta cae se pierde automáticamente las vidas.	

Tarea de Ingeniería	
Número de Tarea: T12-01	Historias de Usuario (Nro. y Nombre): US12- Jugador
Nombre Tarea: Perder Vidas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 04 de Febrero de 2013	Fecha Fin: 04 de Febrero de 2013
Desarrollador Responsable: Gabriel Armendáriz	
Descripción: Cuando la bola cae el grafico que representa las vidas desaparece uno a uno.	

Tarea de Ingeniería	
Número de Tarea: T14-01	Historias de Usuario (Nro. y Nombre): US14- Diseñador
Nombre Tarea: Fondo musical	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 04 de Febrero de 2013	Fecha Fin: 04 de Febrero de 2013
Desarrollador Responsable: Milton Saltos	
Descripción: Cuando aparece la pantalla del menú se escucha una música para hacer agradable el juego.	

CRC de los User Story Primera Iteración

Nombre de la Clase: Bloques	
Responsabilidades: Las responsabilidades de esta clase son: Esquema.	Colaboradores: La Clase Bitmap
Atributos: <code>private</code> Bitmap bloque; <code>private</code> int[][] matriz;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Update() actualizar la bola cuando se está posicionando en la pantalla redimensionaBloques() es un método en el cual se da tamaño a los bloques, número de filas y columnas.	

Nombre de la Clase: Bola	
Responsabilidades: Las responsabilidades de esta clase son: Movimiento. Colisión. Rebotar.	Colaboradores: La Clase Bitmap, pala, bloques, GameView, GameThread.
Atributos: <code>private</code> Bitmap bola; <code>private</code> Bloques bloques; <code>private</code> int pos_x; <code>private</code> int pos_y; <code>private</code> Pala pala; <code>private</code> int v_x = 8; <code>private</code> int v_y = 8;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Update() actualizar la bola cuando se está posicionando en la pantalla.	

Nombre de la Clase: Pala	
Responsabilidades: Las responsabilidades de esta clase son: Mover. Colisión. Actualización.	Colaboradores: Tenemos las clases como son: Bloques, Bola, GameView, GameThread.
Atributos: <code>private</code> Bitmap pala; <code>private</code> int pos_x; <code>private</code> int pos_y; <code>private</code> int width;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Oncreate() es el constructor de la clase. setPosition () evento en el cual se va hallando las posiciones de la pala. Colision() método en el cual se hace rebotar la bola en la pala. onClick() evento del botón mediante el cual reacciona al dar un clic. update () para realizar el movimiento de la pala e ir actualizando constantemente. onDestroy() para destruir una clase.	

Nombre de la Clase: GameView	
Responsabilidades: Las responsabilidades de esta clase son: Mover. Actualizar. Contar Perdidas.	Colaboradores: Tenemos las clases como son: Bola, GamaThread, Pala, Bloques.
Atributos: vidas = 3; protected int width, height, acelerometro, vidas; private GameThread thread; private Pala pala; private Bloques bloques; private Bola bola; private boolean jugando;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. onDraw () para ir dibujando la bola y la pala e ir disminuir las vidas surfaceCreate() para ir creando la onAccuracyChanged () es un método propio del surfaceView necesario que solo este codificado onTouchEvent () evento mediante el cual funciona la pantalla táctil. surfFaceDestroy() para destruir una clase.	

Nombre de la Clase: GameView	
Responsabilidades: Las responsabilidades de esta clase son: Mover. Actualizar. Finalizar Juego.	Colaboradores: Tenemos las clases como son: Bola, GamaThread, Pala, Bloques.
Atributos: vidas = 3; protected int width, height, acelerometro, vidas; private GameThread thread; private Pala pala; private Bloques bloques; private Bola bola; private boolean jugando;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. onDraw () para ir dibujando la bola y la pala e ir disminuir las vidas surfaceCreate() para ir creando la onAccuracyChanged () es un método propio del surfaceView necesario que solo este codificado onTouchEvent () evento mediante el cual funciona la pantalla táctil. surfFaceDestroy() para destruir una clase.	

CRC de los User Story Segunda Iteración

Nombre de la Clase: Instrucciones	
Responsabilidades: Las responsabilidades de esta clase son: Navegar entre pantallas. Ver Instrucciones.	Colaboradores: Tenemos el archivo xml activity_instrucciones, imágenes, tipo de letras.
Atributos: <code>private</code> TextView <code>txt_titulo</code> ; <code>private</code> Button <code>btn_regresar</code> ;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Oncreate() es el constructor de la clase. <code>setOnClickListener()</code> para eventos del botón. <code>onClick()</code> evento del botón mediante el cual reacciona al dar un clic. <code>startActivity()</code> para navegar entre actividades y hacer llamadas a las mismas. <code>onDestroy()</code> para destruir una clase.	

Nombre de la Clase: créditos	
Responsabilidades: Las responsabilidades de esta clase son: Navegar entre pantallas. Ver Créditos.	Colaboradores: Tenemos el archivo xml activity_credits, imágenes, tipo de letras.
Atributos: <code>private</code> TextView <code>txt_titulo</code> ; <code>private</code> Button <code>btn_regresar</code> ;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Oncreate() es el constructor de la clase. <code>setOnClickListener()</code> para eventos del botón. <code>onClick()</code> evento del botón mediante el cual reacciona al dar un clic. <code>startActivity()</code> para navegar entre actividades y hacer llamadas a las mismas. <code>onDestroy()</code> para destruir una clase.	

Nombre de la Clase: PrincipalActivity	
Responsabilidades: Las responsabilidades de esta clase son: Navegar entre pantallas. Iniciar juego. Ver Créditos e Instrucciones.	Colaboradores: Tenemos los archivos xml como son: activity_principal, activity_credits, activity_instrucciones. Asi como también imágenes, tipo de letras.
Atributos: ProgressDialog <code>progressBar</code> ; <code>private int</code> <code>progressBarStatus</code> = 0; <code>private</code> Handler <code>progressBarHandler</code> = new Handler(); <code>private long</code> <code>juegoSize</code> = 0; <code>protected</code> TextView <code>customFont</code> ; <code>private</code> Button <code>btnStartProgress</code> ;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Oncreate() es el constructor de la clase. <code>setOnClickListener()</code> para eventos del botón. <code>onClick()</code> evento del botón mediante el cual reacciona al dar un clic. <code>startActivity()</code> para navegar entre actividades y hacer llamadas a las mismas. <code>onDestroy()</code> para destruir una clase	

Nombre de la Clase: MainActivity	
Responsabilidades: Las responsabilidades de esta clase son: Navegar entre pantallas. Ir al menú principal.	Colaboradores: Tenemos el archivo xml activity_credits, activity_instrucciones, imágenes, tipo de letras.
Atributos: <code>private</code> TextView <code>txt_titulo</code> ; <code>private</code> Button <code>btn_instrucciones</code> ; <code>private</code> Button <code>btn_credits</code> ; <code>private</code> Button <code>btn_salir</code> ;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Oncreate() es el constructor de la clase. setOnClickListener() para eventos del botón. onClick() evento del botón mediante el cual reacciona al dar un clic. startActivity() para navegar entre actividades y hacer llamadas a las mismas. onDestroy() para destruir una clase	

Nombre de la Clase: Bola	
Responsabilidades: Las responsabilidades de esta clase son: Estilo de bola. Posicionamiento de la bola.	Colaboradores:
Atributos: <code>private</code> Bitmap <code>bola</code> ;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Update() actualizar la bola cuando se está posicionando en la pantalla Bola() es un método en el cual se da el aspecto de balón de fútbol a la bola	

Nombre de la Clase: Bloques	
Responsabilidades: Las responsabilidades de esta clase son: Colisión. Actualización. Reproducir Sonido.	Colaboradores: Tenemos las clases como son: Pala, Bola, GameView, GamaThread.
Atributos: <code>private static</code> Bitmap <code>bloque, vida</code> ; <code>private boolean</code> <code>acaba</code> ; <code>private</code> MediaPlayer <code>reproductor</code> ; <code>private</code> MediaPlayer <code>reproductorm</code> ; <code>private</code> String <code>marcador = "PUNTUACION: 0"</code> ; <code>private int</code> <code>puntuacion = 0</code> ; <code>private</code> Paint <code>estiloTexto = new Paint()</code> ; <code>private int</code> <code>col = 16</code> ; <code>private int</code> <code>row = 10</code> ; <code>private int i, j, alto, ancho, width, height</code> ; <code>private int[][] matriz</code>	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Bloques () es el constructor de la clase. Reproducir () es el método mediante el cual se va a reproducir el sonido DimensionBloque () método en el cual se dimensiona el bloque an ancho y alto. Colision() método en el cual desaparece el bloque que fue impactado por la bola. update () para realizar el movimiento de la pala e ir actualizando constantemente. onDestroy() para destruir una clase.	

Nombre de la Clase: Bloques	
Responsabilidades: Las responsabilidades de esta clase son: Colisión. Actualización. Puntuacion..	Colaboradores: Tenemos las clases como son: Pala, Bola, GamaThread, GameView.
Atributos: <code>private static</code> Bitmap <i>bloque, vida</i> ; <code>private boolean</code> <i>acaba</i> ; <code>private</code> MediaPlayer <i>reproductor</i> ; <code>private</code> MediaPlayer <i>reproductorm</i> ; <code>private</code> String <i>marcador = "PUNTUACION: 0"</i> ; <code>private int</code> <i>puntuacion = 0</i> ; <code>private</code> Paint <i>estiloTexto = new</i> Paint(); <code>private int</code> <i>col = 16</i> ; <code>private int</code> <i>row = 10</i> ; <code>private int i, j, alto, ancho, width, height</code> ; <code>private int[][]</code> <i>matriz</i>	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. Bloques () es el constructor de la clase. Reproducir () es el método mediante el cual se va a reproducir el sonido DimensionBloque () método en el cual se dimensiona el bloque an ancho y alto. Colision() método en el cual desaparece el bloque que fue impactado por la bola. update () para realizar el movimiento de la pala e ir actualizando constantemente. onDestroy() para destruir una clase.	

Nombre de la Clase: GameView	
Responsabilidades: Las responsabilidades de esta clase son: Mover. Actualizar. Contar Perdidas.	Colaboradores: Tenemos las clases como son: Bola, GamaThread, Pala, Bloques.
Atributos: <code>vidas = 3</code> ; <code>protected int</code> <i>width, height, acelerometro, vidas</i> ; <code>private</code> GameThread <i>thread</i> ; <code>private</code> Pala <i>pala</i> ; <code>private</code> Bloques <i>bloques</i> ; <code>private</code> Bola <i>bola</i> ; <code>private boolean</code> <i>jugando</i> ;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. onDraw () para ir dibujando la bola y la pala e ir disminuir las vidas surfaceCreate() para ir creando la onAccuracyChanged () es un método propio del surfaceView necesario que solo este codificado onTouchEvent () evento mediante el cual funciona la pantalla táctil. surfFaceDestroy() para destruir una clase.	

Nombre de la Clase: GameView	
Responsabilidades: Las responsabilidades de esta clase son: Mover. Actualizar. Contar Perdidas.	Colaboradores: Tenemos las clases como son: Bola, GamaThread, Pala, Bloques.
Atributos: vidas = 3; protected int width, height, acelerometro, vidas; private GameThread thread; private Pala pala; private Bloques bloques; private Bola bola; private boolean jugando;	
Operaciones: Las operaciones son múltiples entre las cuales tenemos. onDraw () para ir dibujando la bola y la pala e ir disminuir las vidas surfaceCreate() para ir creando la onAccuracyChanged () es un método propio del surfaceView necesario que solo este codificado onTouchEvent () evento mediante el cual funciona la pantalla táctil. surfFaceDestroy() para destruir una clase.	

ANEXO II

PRUEBAS DE ACEPTACIÓN Y TARJETAS DE TAREA DEL VIDEOJUEGO “EL TREN DEL CONOCIMIENTO”.

Pruebas de Aceptación

Prueba de Aceptación	
<i>Identificador: PA01</i>	<i>Historia de Usuario (Nro. Y Nombre): HU01-Como jugador quiero tener un menú principal.</i>
<i>Nombre: Realizar el menú principal.</i>	
<i>Descripción: Realizar un menú principal en donde se pueda escoger Jugar, Instrucciones, Créditos, Puntuaciones y Salir</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil en completo funcionamiento.</i>	
<i>Entrada / Pasos de ejecución:</i> <ol style="list-style-type: none">1. Presentar el menú principal de color rojo degradado con sus respectivas opciones.	
<i>Resultado Esperado: Mostrar las diferentes opciones (pantallas) al dar clic sobre una de ellas.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Prueba de Aceptación	
<i>Identificador: PA02</i>	<i>Historia de Usuario (Nro. Y Nombre): HU02- Como jugador quiero que se muestre un Splash al iniciar la aplicación (videojuego).</i>
<i>Nombre: Mostrar Splash.</i>	
<i>Descripción: Mostrar un Splash al iniciar el juego.</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil con todos los efectos en completo funcionamiento.</i>	
<i>Entrada / Pasos de ejecución:</i> <ol style="list-style-type: none">1. Ingresar al juego.2. Mostrar el Splash	
<i>Resultado Esperado: Mostrar el Splash con todos sus efectos.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Prueba de Aceptación	
Identificador: PA03	Historia de Usuario (Nro. Y Nombre): HU03- <i>Como jugador quiero que los objetos aparezcan en la vía del tren.</i>
Nombre: Los objetos aparezcan en la vía del tren	
Descripción: Los objetos vayan apareciendo únicamente en la vía del tren para irlos cogiéndolos.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal escoger Jugar. 3. Pulsar sobre el nivel que se encuentre deshabilitado 4. Para iniciar el juego pulsar en cualquier parte de la pantalla 5. Los objetos van apareciendo según el tren los vaya cogiendo 	
Resultado Esperado: Los objetos no deben aparecer fuera de la vía del tren.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA04	Historia de Usuario (Nro. Y Nombre): HU04- <i>Como jugador quiero visualizar las Instrucciones del juego</i>
Nombre: Visualizar las instrucciones del juego.	
Descripción: Visualizar las instrucciones del juego para saber cómo jugar	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Instrucciones. 3. Visualizar las instrucciones del juego 	
Resultado Esperado: Visualizar que las instrucciones sean claras y que ayuden a tener una mejor visión de cómo poder jugar.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA05	Historia de Usuario (Nro. Y Nombre): HU05- <i>Como jugador quiero visualizar los créditos del juego</i>
Nombre: Visualizar los créditos del juego.	
Descripción: Visualizar los créditos del juego para saber quién lo realizó.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Créditos. 3. Visualizar los créditos del juego 	
Resultado Esperado: Visualizar los créditos del juego para saber quién o quienes lo realizaron.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
<i>Identificador: PA06</i>	<i>Historia de Usuario (Nro. Y Nombre): HU06- Como jugador quiero visualizar las 10 más altas puntuaciones.</i>
<i>Nombre: Visualizar las 10 puntuaciones más altas.</i>	
<i>Descripción: Visualizar las 10 puntuaciones más altas del juego para saber cuáles jugadores la sacaron.</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.</i>	
<i>Entrada / Pasos de ejecución:</i>	
<ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Puntuaciones. 3. Visualizar la lista de jugadores en caso de haber sido previamente guardados. 	
<i>Resultado Esperado: Visualizar las 10 puntuaciones más altas de los jugadores conjuntamente con sus nombres y preguntas correctas e incorrectas.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Prueba de Aceptación	
<i>Identificador: PA07</i>	<i>Historia de Usuario (Nro. Y Nombre): HU07- Como jugador quiero que los controles del tren se encuentren en la parte inferior de la pantalla distribuidos en los extremos respectivamente.</i>
<i>Nombre: Controles del tren.</i>	
<i>Descripción: Los controles del tren se encuentran en la parte inferior de la pantalla distribuidos en los extremos respectivamente para que pueda facilitar la jugabilidad.</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.</i>	
<i>Entrada / Pasos de ejecución:</i>	
<ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre deshabilitado 4. Pulsar sobre la pantalla para iniciar el juego 5. Visualizar controles del tren en la parte inferior de la pantalla 	
<i>Resultado Esperado: Visualizar que los controles del tren en la parte inferior de la pantalla distribuidos en los extremos respectivamente.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Prueba de Aceptación	
<i>Identificador: PA08</i>	<i>Historia de Usuario (Nro. Y Nombre): HU08- Como jugador quiero que el tren pueda moverse solo por los rieles o caminos.</i>
<i>Nombre: Mover el tren por sus rieles o caminos.</i>	
<i>Descripción: El tren se mueva solo por los rieles sin desviarse del mismo para no terminar el juego.</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.</i>	
<i>Entrada / Pasos de ejecución:</i>	
<ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar en un nivel que se encuentre deshabilitado 4. Pulsar sobre la pantalla para iniciar el juego 5. Tomar el control del tren y llevarlo por el camino correcto 	
<i>Resultado Esperado: Que el tren se pueda mover libremente por todos sus rieles o caminos y si se sale de uno de ellos pierda el juego.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Prueba de Aceptación	
Identificador: PA09	Historia de Usuario (Nro. Y Nombre): HU09- <i>Como jugador quiero que el tren pueda girar más de 90 grados con menos de 2 vagones</i>
Nombre: <i>El tren pueda girar más de 90 grados cuando tenga un vagón</i>	
Descripción: <i>El tren pueda girar más de 90 grados con menos de 2 vagones para que pueda darse la vuelta cuando cambia de rumbo.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre un nivel que se encuentre deshabilitado 4. Pulsar sobre la pantalla para iniciar el juego 5. Gira el tren más de 90 grados solo cuando este tenga menos de 2 vagones 	
Resultado Esperado: <i>Quiero que el tren gire más de 90 grados cuando tenga un solo vagón.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA10	Historia de Usuario (Nro. Y Nombre): HU10- <i>Como jugador quiero que el tren no pueda girar más de 90 grados</i>
Nombre: <i>El tren no pueda girar más de 90 grados.</i>	
Descripción: <i>El tren no pueda girar más de 90 grados ya que si lo hace chocara con sus vagones y terminará el juego.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre un nivel que se encuentre deshabilitado 4. Pulsar sobre la pantalla para iniciar el juego 5. Gira el tren 90 grados cuando este tenga más de 1 vagón 	
Resultado Esperado: <i>Quiero que el tren cuando gire más de 90 grados y tenga más de 1 vagón pierda.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA11	Historia de Usuario (Nro. Y Nombre): HU11- <i>Como jugador que el tren al coger los objetos se escuche un sonido de "incremento".</i>
Nombre: <i>Escuchar sonido cuando el tren coja objetos.</i>	
Descripción: <i>El tren al coger objetos se escuche un sonido de incremento para dar realismo al juego.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil y el sonido debe escucharse claramente.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre un nivel que se encuentre deshabilitado 4. Pulsar sobre la pantalla para iniciar el juego 5. Guiar al tren hacia el objeto y pasar por el para cogerlo. 	
Resultado Esperado: <i>Cuando el tren coja los objetos que van apareciendo en los rieles se escuche un sonido.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA12	Historia de Usuario (Nro. Y Nombre): HU12- Como jugador quiero que existan 8 niveles los mismos que se encuentren deshabilitados exceptuando el primero.
Nombre: Exista 8 niveles deshabilitados excepto el primero	
Descripción: Existan 8 niveles en el juego los mismos que se encuentren deshabilitados exceptuando el primero para al ir completando un nivel se vaya deshabilitando el siguiente	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil y los 8 niveles deben aparecer deshabilitados excepto el primero	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Visualizar que los niveles se encuentren deshabilitados menos el primero 	
Resultado Esperado: Que exista una pantalla de 8 niveles los mismos que se encuentren deshabilitados mientras no hayan sido superados a excepción del primero.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA13	Historia de Usuario (Nro. Y Nombre): HU13- Como jugador quiero ingresar mi nombre y apellido y una foto si lo deseara
Nombre: Ingreso de datos personales	
Descripción: Ingresar mi nombre y apellido y foto (opcional) para visualizar junto con la puntuación alcanzada	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil y guardar la información en la base de datos SQLite	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre habilitado 4. Cuando el tren se salga de su vía pierde. 5. Pulsar sobre salir. 6. Agregar foto(opcional) 7. Ingresar nombre 8. Ingresar apellido 9. Pulsar sobre Guardar. 	
Resultado Esperado: Visualizar en la pantalla puntuaciones todos los datos del jugador ingresados conjuntamente con su puntuación.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA14	Historia de Usuario (Nro. Y Nombre): HU14- Como jugador quiero visualizar el puntaje al coger los objetos y al responder correctamente las preguntas.
Nombre: Visualizar el puntaje al coger los objetos y al responder correctamente las preguntas.	
Descripción: Visualizar el puntaje cuando el tren vaya cogiendo los objetos y respondiendo las preguntas que irán apareciendo.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre habilitado 4. Coger objetos que aparecen en la vía del tren 5. Responder las preguntas 6. Visualizar la puntuación que se va alcanzando 	
Resultado Esperado: Que cuando el tren coja los objetos y responda las preguntas la puntuación se visualice la puntuación que se va alcanzando.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
<i>Identificador: PA15</i>	<i>Historia de Usuario (Nro. Y Nombre): HU15- Como jugador quiero pausar el juego al pulsar un botón</i>
<i>Nombre: Pausar el juego pulsando un botón</i>	
<i>Descripción: Pausar el juego pulsando un botón para poder hacer algo en ese momento o salir del juego.</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.</i>	
<i>Entrada / Pasos de ejecución:</i>	
<ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre habilitado 4. Coger objetos que aparecen en la vía del tren 5. Presionar el botón de pausado 6. Presionar en Continuar o Salir 	
<i>Resultado Esperado: Cuando se pulse el botón de pausado el juego se detenga y permita continuar o salir del mismo.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Prueba de Aceptación	
<i>Identificador: PA16</i>	<i>Historia de Usuario (Nro. Y Nombre): HU16- Como jugador quiero cancelar ingreso de datos</i>
<i>Nombre: Cancelar ingreso de datos</i>	
<i>Descripción: Cancelar el ingreso de datos personales de modo que no pueda visualizar mi puntuación.</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.</i>	
<i>Entrada / Pasos de ejecución:</i>	
<ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre habilitado 4. Coger objetos que aparecen en la vía del tren 5. Si el tren se sale de su vía o choca contra sus vagones 6. Fin del juego 7. Pulsar en salir 8. Pulsar en Cancelar. 	
<i>Resultado Esperado: Cuando se pulse en el botón cancelar de la pantalla de ingreso de datos pueda regresar al menú principal y no se almacene ninguna información.</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Prueba de Aceptación	
<i>Identificador: PA17</i>	<i>Historia de Usuario (Nro. Y Nombre): HU17- Como jugador quiero que El tren aumente los vagones a medida que vaya cogiendo 3 objetos.</i>
<i>Nombre: El tren aumente de vagones cada que coja 3 objetos</i>	
<i>Descripción: El tren aumente los vagones a medida que vaya cogiendo 3 objetos, para aumentar la diversión del juego</i>	
<i>Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil y debe aparecer los vagones.</i>	
<i>Entrada / Pasos de ejecución:</i>	
<ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre habilitado 4. Coger objetos que aparecen en la vía del tren 5. Cuando el tren haya cogido 3 objetos 6. Visualizar incremento de vagones. 	
<i>Resultado Esperado: Cuando el tren coja 3 objetos aumente un vagón</i>	
<i>Evaluación de la Prueba: Correcto</i>	

Prueba de Aceptación	
Identificador: PA18	Historia de Usuario (Nro. Y Nombre): HU18- <i>Como jugador quiero que las preguntas vayan apareciendo a medida que vaya cogiendo 6 objetos.</i>
Nombre: <i>Las preguntas vayan apareciendo cuando se coa 6 objetos</i>	
Descripción: <i>Las preguntas vayan apareciendo a medida que vaya cogiendo 6 objetos para aumentar la diversión del juego.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre habilitado 4. Coger los objetos que van apareciendo en la vía del tren 5. Cuando el tren haya cogido 6 objetos 6. Visualizar la pregunta 	
Resultado Esperado: <i>Visualizar una pregunta cada vez que el tren haya cogido 6 objetos.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA19	Historia de Usuario (Nro. Y Nombre): HU19- <i>Como jugador quiero que antes empezar a jugar se visualice el nivel del juego en una pantalla de preparado</i>
Nombre: <i>Visualizar el nivel del juego en una pantalla preparado</i>	
Descripción: <i>Antes empezar a jugar se visualice el nivel del juego en una pantalla de preparado para que sea más óptimo.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre habilitado 4. Visualizar la pantalla de preparación del juego o del nivel 	
Resultado Esperado: <i>Antes de iniciar a jugar el nivel aparezca la pantalla de preparación del mismo.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA20	Historia de Usuario (Nro. Y Nombre): HU20- <i>Como jugador quiero pulsar en cualquier parte de la pantalla preparado</i>
Nombre: <i>Pulsar en cualquier parte de la pantalla preparado</i>	
Descripción: <i>Pulsar en cualquier parte de la pantalla preparado, para empezar a jugar.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre el nivel que se encuentre habilitado 4. Visualizar la pantalla de preparación del nivel 5. Pulsar en cualquier parte de la pantalla para jugar 	
Resultado Esperado: <i>Al pulsar en cualquier parte de la pantalla preparado empezar a jugar.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA21	Historia de Usuario (Nro. Y Nombre): HU21- <i>Como jugador quiero visualizar una pantalla de "Fin del Juego" al perder el juego conjuntamente con dos botones de "Nuevo Juego" y "Salir"</i>
Nombre: Visualizar una pantalla de fin de juego.	
Descripción: Visualizar una pantalla de "Fin del Juego" al perder el juego conjuntamente con dos botones de "Nuevo Juego" y "Salir" para saber que perdí, de modo que pueda intentarlo de nuevo o salir del juego.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre un nivel habilitado 4. Jugar 5. Cuando pierde aparece una pantalla de salida 6. Pulsar en continuar o salir. 	
Resultado Esperado: Al perder el juego poder jugar nuevamente el nivel o salir del mismo.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA22	Historia de Usuario (Nro. Y Nombre): HU22 - <i>Como jugador quiero poder responder las preguntas que se van mostrando mediante dos botones de afirmación (SI) o negación (NO).</i>
Nombre: Responder a las preguntas que se van mostrando mediante botones de si o no	
Descripción: Poder responder las preguntas que se van mostrando mediante dos botones de afirmación (SI) o negación (NO) para saber el grado de conocimiento que tengo sobre la Provincia de Chimborazo y adquirir nuevo conocimiento.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar sobre Jugar. 3. Pulsar sobre un nivel habilitado 4. Pulsar sobre la pantalla preparado 5. Coger los objetos y esperar que salga una pregunta 6. Visualizar los botones de respuesta afirmativa o negativa 	
Resultado Esperado: Cuando el tren haya cogido 6 objetos aparezca la pregunta y tenga opciones de respuesta afirmativa (SI) o negativa (NO).	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA23	Historia de Usuario (Nro. Y Nombre): HU23- <i>Como jugador quiero que en cada nivel se muestren como máximo 5 preguntas.</i>
Nombre: Cada nivel muestre un máximo de 5 preguntas	
Descripción: En cada nivel se muestren como máximo 5 preguntas para que el juego no sea aburrido.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal pulsar Jugar. 3. Pulsar sobre un nivel habilitado 4. Pulsar sobre la pantalla preparado 5. Coger los objetos y esperar que salga una pregunta 6. Coger un máximo de 5 preguntas 	
Resultado Esperado: Cuando haya respondido 5 preguntas el nivel será superado.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA24	Historia de Usuario (Nro. Y Nombre): HU24- <i>Como jugador quiero que al escoger en el menú principal la opción salir se muestre una pantalla de confirmación de cerrar la aplicación.</i>
Nombre: Cerrar la aplicación (videojuego).	
Descripción: Al escoger en el menú principal la opción salir se muestre una pantalla de confirmación de cerrar la aplicación en caso de estar seguro hacerlo o de lo contrario ir al menú principal nuevamente.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar sobre Salir del menú principal 3. Aparecerá un pantalla de confirmación de salir de la aplicación 4. Pulsar en cualquiera de las opciones SI o NO. 	
Resultado Esperado: Cuando pulse sobre la opción Salir del menú principal visualice una pantalla de salida en donde tenga opciones de afirmación (SI) o negación (NO).	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA25	Historia de Usuario (Nro. Y Nombre): HU25- <i>Como jugador quiero que existan 2 botones para poder acceder a las redes sociales del Facebook o Twitter.</i>
Nombre: Exista 2 botones para acceder a las redes sociales.	
Descripción: Existan 2 botones para poder acceder a las redes sociales del Facebook o Twitter para conocer opiniones del juego conocer información adicional del mismo.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil y debe acceder a las redes sociales como facebook y twitter.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. En el menú principal visualizar los accesos a las redes sociales como Facebook y Twitter 	
Resultado Esperado: Visualizar los iconos de las redes sociales como facebook y twitter y acceder sus páginas respectivas sin ningún problema.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA26	Historia de Usuario (Nro. Y Nombre): HU26- <i>Como jugador quiero que exista un botón de sonido en la pantalla de menú principal.</i>
Nombre: Botón de sonido en la pantalla del menú principal	
Descripción: Exista un botón de sonido en la pantalla de menú principal para que se pueda habilitar y deshabilitar el sonido.	
Condiciones de Ejecución: Debe ejecutarse en el dispositivo móvil.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Visualizar el botón de sonido en la pantalla del menú principal 3. Click sobre el mismo 4. Se deshabilitará el sonido 	
Resultado Esperado: Al pulsar sobre el botón del sonido del menú principal se deshabilita el mismo y luego se habilita de la misma forma.	
Evaluación de la Prueba: Correcto	

Prueba de Aceptación	
Identificador: PA27	Historia de Usuario (Nro. Y Nombre): HU27- <i>Como sonidista quiero que exista sonido cuando el tren este en movimiento</i>
Nombre: <i>Exista sonido cuando el tren esté en movimiento</i>	
Descripción: <i>Exista sonido cuando el tren este en movimiento para dar un aspecto real al juego.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en la opción del menú Jugar 3. Pulsar sobre un nivel habilitado 4. Pulsar sobre la pantalla de preparación del juego 5. Escuchar sonido del tren cuando este en movimiento 	
Resultado Esperado: <i>Cuando el tren este en movimiento escuchar el sonido que este produce.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA28	Historia de Usuario (Nro. Y Nombre): HU28- <i>Como diseñador quiero que el tren con aspecto real.</i>
Nombre: <i>El tren tenga un aspecto real.</i>	
Descripción: <i>El tren tenga un aspecto real, de modo que se muestre más divertido</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en la opción del menú Jugar 3. Pulsar sobre un nivel habilitado 4. Pulsar sobre la pantalla de preparación del juego 5. Visualizar el aspecto del tren 	
Resultado Esperado: <i>Visualizar que el tren tenga un aspecto real.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA29	Historia de Usuario (Nro. Y Nombre): HU29- <i>Como diseñador quiero que los vagones tengan un aspecto de un cuadrado de color rojo.</i>
Nombre: <i>Los vagones tengan aspecto de un cuadrado de color rojo</i>	
Descripción: <i>Los vagones tengan un aspecto de un cuadrado de color rojo para que se muestre realista</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en la opción del menú Jugar 3. Pulsar sobre un nivel habilitado 4. Pulsar sobre la pantalla de preparación del juego 5. Coger 3 objetos para aumentar un vagón 6. Visualizar que el vagón sea cuadrado y de color rojo 	
Resultado Esperado: <i>Visualizar que el vagón del tren tenga un aspecto cuadrado de color rojo.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA30	Historia de Usuario (Nro. Y Nombre): HU30- <i>Como diseñador quiero que los objetos tengan un aspecto de cofres, hongos y monedas de colores dorados y cafés.</i>
Nombre: <i>Objetos con aspecto de cofres, hongos y monedas</i>	
Descripción: <i>Los objetos tengan un aspecto de cofres, hongos y monedas de colores dorados y cafés para que se muestren divertidos.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en la opción del menú Jugar 3. Pulsar sobre un nivel habilitado 4. Pulsar sobre la pantalla de preparación del juego 5. Visualizar el aspecto de los objetos que son cofres, monedas y hongos 	
Resultado Esperado: <i>Visualizar que el aspecto de los objetos que aparecen sean cofres, hongos y monedas.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA31	Historia de Usuario (Nro. Y Nombre): HU31- <i>Como diseñador quiero que existan animaciones en la pantalla del menú principal, créditos, instrucciones, niveles y salir</i>
Nombre: <i>Animaciones en las pantallas</i>	
Descripción: <i>Existan animaciones en la pantalla del menú principal, créditos, instrucciones, niveles y salir para dar un aspecto divertido.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Visualizar animaciones en la pantalla del menú principal 3. Visualizar animaciones en la pantalla de Créditos 4. Visualizar animaciones en la pantalla de Instrucciones 5. Visualizar animaciones en la pantalla de Niveles 6. Visualizar animaciones en la pantalla de Salir 	
Resultado Esperado: <i>Visualizar animaciones en las pantallas de menú principal, créditos, instrucciones, niveles y salir.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA32	Historia de Usuario (Nro. Y Nombre): HU32- <i>Como diseñador quiero que exista en la pantalla de créditos, instrucciones, puntuaciones, niveles y salir un botón de regreso.</i>
Nombre: <i>Botón regreso al menú principal</i>	
Descripción: <i>Exista en la pantalla de créditos, instrucciones, puntuaciones, niveles y salir un botón de regreso Para poder volver al menú principal</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en la opción del menú principal Jugar 3. Pulsar en el botón salir de la pantalla de niveles 4. Pulsar en la opción del menú principal Créditos 5. Pulsar en el botón salir de la pantalla de créditos 6. Pulsar en la opción del menú principal Instrucciones 7. Pulsar en el botón salir de la pantalla instrucciones 8. Pulsar en la opción del menú principal Puntuaciones 9. Pulsar en el botón salir de la pantalla Puntuaciones 10. Pulsar en la opción del menú principal Salir 11. Pulsar en el botón no 	
Resultado Esperado: <i>Al pulsar en los salir de las diferentes pantallas como créditos, instrucciones, puntuaciones, niveles y salir regresar al menú principal</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA33	Historia de Usuario (Nro. Y Nombre): HU33- <i>Como Sonidista quiero que exista audio de fondo en las diferentes pantallas del videojuego.</i>
Nombre: <i>Exista audio de fondo en diferentes pantallas</i>	
Descripción: <i>Exista audio de fondo en las diferentes pantallas del videojuego para que sea más divertido</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Escuchar audio en la pantalla del menú principal 3. Pulsar sobre la opción del menú principal Jugar 4. Escuchar audio en la pantalla de niveles 5. Pulsar sobre la opción del menú principal Instrucciones 6. Escuchar audio en la pantalla de Instrucciones 7. Pulsar sobre la opción del menú principal Créditos 8. Escuchar audio en la pantalla de créditos 9. Pulsar sobre la opción del menú principal Puntuaciones 10. Escuchar audio en la pantalla de puntuaciones 11. Pulsar sobre la opción del menú principal Salir 12. Escuchar audio en la pantalla de salir 	
Resultado Esperado: <i>Al estar en las diferentes pantallas del menú principal de debe escuchar un audio de fondo.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA34	Historia de Usuario (Nro. Y Nombre): HU34- <i>Como diseñador quiero que la pantalla de puntuaciones muestre el nombre, apellido, puntuación, número de preguntas correctas e incorrectas contestadas de los 10 jugadores con más altas puntuaciones.</i>
Nombre: <i>Mostrar en la pantalla puntuaciones algunos campos en la lista</i>	
Descripción: <i>La pantalla de puntuaciones muestre el nombre, apellido, puntuación, número de preguntas correctas e incorrectas contestadas de los 10 jugadores con más altas puntuaciones para que se pueda apreciar mejor</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar sobre Puntuaciones del menú principal 3. Visualizar los datos de los jugadores como nombre, apellido, puntuación, preguntas correctas y preguntas incorrectas 	
Resultado Esperado: <i>Visualizar a detalle completo los datos de los jugadores</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA35	Historia de Usuario (Nro. Y Nombre): HU35- <i>Como diseñador quiero que exista en la pantalla de Instrucciones botones de navegabilidad entre sus pantallas.</i>
Nombre: <i>La pantalla de Instrucciones tenga botones de navegabilidad</i>	
Descripción: <i>Exista en la pantalla de Instrucciones botones de navegabilidad para interactuar fácilmente</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar sobre Instrucciones del menú principal 3. Visualizar botones hacia adelante o atrás 4. Pulsar en los botones y navegar en las pantallas de instrucciones. 	
Resultado Esperado: <i>Cuando ingrese a instrucciones del menú principal poder y navegar fácilmente por todas las instrucciones del juego.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA36	Historia de Usuario (Nro. Y Nombre): HU36- <i>Como sonidista quiero que se escuche un sonido al iniciarse el Splash.</i>
Nombre: <i>Sonido en el Splash</i>	
Descripción: <i>Se escuche un sonido al iniciarse el Splash para que sea divertido.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Escuchar sonido en el Splash 	
Resultado Esperado: <i>Cuando ingrese al juego escuchar un sonido cuando aparezca el Splash.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA37	Historia de Usuario (Nro. Y Nombre): HU37- <i>Como jugador quiero que el contenido de las preguntas sean sobre la provincia de Chimborazo</i>
Nombre: <i>Contenido de las preguntas sean sobre la provincia de Chimborazo</i>	
Descripción: <i>El contenido de las preguntas sean sobre la provincia de Chimborazo para complementar el conocimiento</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en Jugar del menú principal 3. Pulsar sobre un nivel habilitado 4. Pulsar en cualquier parte de la pantalla para iniciar a jugar 5. Con el tren coger 5 objetos 6. Aparece la pregunta la misma que en su contenido debe presentar información sobre la provincia de Chimborazo 	
Resultado Esperado: <i>Todas las preguntas estén bien formuladas entendibles y que en su contenido sean exclusivo de la provincia de Chimborazo.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación

Identificador: PA38	Historia de Usuario (Nro. Y Nombre): HU38- <i>Como Diseñador quiero que al responderse una pregunta correctamente se muestre un mensaje de "CORRECTO" con un visto de color verde, con un botón de "continuar" para seguir jugando, conjuntamente con un sonido de correcto.</i>
Nombre: <i>Al responderse una pregunta correctamente se muestre un mensaje de "CORRECTO".</i>	
Descripción: <i>Al responderse una pregunta correctamente se muestre un mensaje de "CORRECTO" con un visto de color verde, con un botón de "continuar" para seguir jugando, conjuntamente con un sonido de correcto para saber que se respondió correctamente.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1. Ingresar al juego.2. Pulsar en Jugar del menú principal3. Pulsar sobre un nivel habilitado4. Pulsar en cualquier parte de la pantalla para iniciar a jugar5. Con el tren coger 5 objetos6. Aparece la pregunta la misma que en su contenido debe presentar información sobre la provincia de Chimborazo7. Si la respuesta es bien respondida8. Mostrar un mensaje de correcto	
Resultado Esperado: <i>Cuando se cojan 5 objetos aparezcan las preguntas y al responder correctamente mostrar un mensaje "CORRECTO".</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación

Identificador: PA39	Historia de Usuario (Nro. Y Nombre): HU39- <i>Como Diseñador quiero que al responderse una pregunta incorrectamente se muestre un mensaje de "INCORRECTO" con una x de color rojo, un botón de "continuar" para seguir jugando, conjuntamente con un sonido de incorrecto.</i>
Nombre: <i>Al responderse una pregunta incorrectamente se muestre un mensaje de "INCORRECTO".</i>	
Descripción: <i>Al responderse una pregunta incorrectamente se muestre un mensaje de "INCORRECTO" con una x de color rojo, un botón de "continuar" para seguir jugando, conjuntamente con un sonido de incorrecto para saber que se respondió incorrectamente.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1. Ingresar al juego.2. Pulsar en Jugar del menú principal3. Pulsar sobre un nivel habilitado4. Pulsar en cualquier parte de la pantalla para iniciar a jugar5. Con el tren coger 5 objetos6. Aparece la pregunta la misma que en su contenido debe presentar información sobre la provincia de Chimborazo7. Si la respuesta es incorrecta8. Aparece un mensaje de incorrecto y se escucha un sonido	
Resultado Esperado: <i>Cuando se cojan 5 objetos aparezcan las preguntas y al responder incorrectamente mostrar un mensaje "INCORRECTO" y se escuchara un sonido.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA40	Historia de Usuario (Nro. Y Nombre): HU40- <i>Como jugador quiero que al responder una pregunta incorrectamente se muestre la respuesta correcta.</i>
Nombre: <i>Responder una pregunta incorrecta mostrar lo correcto</i>	
Descripción: <i>Al responder una pregunta incorrectamente se muestre la respuesta correcta para que pueda saber que es lo correcto.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en Jugar del menú principal 3. Pulsar sobre un nivel habilitado 4. Pulsar en cualquier parte de la pantalla para iniciar a jugar 5. Con el tren coger 5 objetos 6. Aparece la pregunta la misma que en su contenido debe presentar información sobre la provincia de Chimborazo 7. Si la respuesta es incorrecta y la pregunta es falsa se mostrara lo correcto de la pregunta 	
Resultado Esperado: <i>Cuando la pregunta sea falsa se visualizara la respuesta correcta.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA41	Historia de Usuario (Nro. Y Nombre): HU41- <i>Como jugador quiero que la puntuación del juego se vaya acumulando nivel tras nivel</i>
Nombre: <i>Puntuación del juego se vaya acumulando</i>	
Descripción: <i>La puntuación del juego se vaya acumulando nivel tras nivel para conocer el total alcanzado al finalizar el juego.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en Jugar del menú principal 3. Pulsar sobre un nivel habilitado 4. Pulsar en cualquier parte de la pantalla para iniciar a jugar 5. Con el tren coger 5 objetos 6. Aparece la pregunta la misma que en su contenido debe presentar información sobre la provincia de Chimborazo 7. Responder a la pregunta 8. Pulsar sobre continuar para seguir jugando 9. Cuando se haya respondido 5 preguntas el nivel es completado 10. La puntuación alcanzada se visualizará en el siguiente nivel 	
Resultado Esperado: <i>Cuando complete los niveles y no se ha perdido el juego la puntuación debe seguirse acumulando.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA42	Historia de Usuario (Nro. Y Nombre): HU42- <i>Como jugador quiero que la puntuación disminuya cuando se responda incorrectamente en 10 puntos</i>
Nombre: <i>Puntuación disminuya en 10 puntos al responder incorrectamente</i>	
Descripción: <i>La puntuación disminuya cuando se responda incorrectamente en 10 puntos para que no pueda alcanzar la máxima puntuación</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en Jugar del menú principal 3. Pulsar sobre un nivel habilitado 4. Pulsar en cualquier parte de la pantalla para iniciar a jugar 5. Coger todos los objetos que van apareciendo 6. Cuando se ha cogido 5 objetos aparecerán las preguntas 7. Si responde incorrectamente 8. Disminuye en 10 puntos a la puntuación total 	
Resultado Esperado: <i>Cuando se responda incorrectamente las preguntas la puntuación debe disminuir 10 puntos.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Prueba de Aceptación	
Identificador: PA43	Historia de Usuario (Nro. Y Nombre): HU43- <i>Como jugador quiero que al completar todos los niveles y al responder correctamente todas las preguntas se muestre una pantalla de congratulación.</i>
Nombre: <i>Mostrar una pantalla de congratulación cuando se hayan completado con éxito todos los niveles</i>	
Descripción: <i>Al completar todos los niveles y al responder correctamente todas las preguntas se muestre una pantalla de congratulación para que pueda sentirse elogiado.</i>	
Condiciones de Ejecución: <i>Debe ejecutarse en el dispositivo móvil.</i>	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al juego. 2. Pulsar en Jugar del menú principal 3. Pulsar sobre un nivel habilitado 4. Pulsar en cualquier parte de la pantalla para iniciar a jugar 5. Coger 5 objetos 6. Aparece la pregunta 7. Si responde 5 preguntas 8. Nivel completado 9. Si completo todos los niveles y todas la preguntas fueron respondidas correctamente 10. Visualizar pantalla de congratulación 	
Resultado Esperado: <i>Cuando se complete todos los niveles y se hayan respondido correctamente las preguntas se debe visualizar una pantalla de congratulación.</i>	
Evaluación de la Prueba: <i>Correcto</i>	

Tarjetas de Tareas Primera Iteración

Tarjeta de Tarea	
<i>Número de Tarea:</i> T02	<i>Historia de Usuario (Nro. Y Nombre):</i> HU07-Programador
<i>Nombre Tarea:</i> Posicionar los controles del tren	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 16 horas
<i>Fecha Inicio:</i> 26 de Marzo de 2013	<i>Fecha Fin:</i> 27 de Marzo de 2013
<i>Miembro Responsable:</i> Gabriel Armendáriz	
<i>Descripción:</i> Los controles del tren deben estar en la parte inferior de la pantalla distribuidos en los extremos de la misma para tener facilidad de jugabilidad.	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T03	<i>Historia de Usuario (Nro. Y Nombre):</i> HU08-Programador
<i>Nombre Tarea:</i> El tren gire más de 90 grados con menos de 2 vagones	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 8 horas
<i>Fecha Inicio:</i> 28 de Marzo de 2013	<i>Fecha Fin:</i> 28 de Marzo de 2013
<i>Miembro Responsable:</i> Milton Saltos	
<i>Descripción:</i> Cuando el tren no tenga menos de 2 vagones podrá girar más de 90 grados de modo que pueda darse la vuelta para cambiar de rumbo	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T04	<i>Historia de Usuario (Nro. Y Nombre):</i> HU09-Programador
<i>Nombre Tarea:</i> El tren no pueda girar más de 90 grados	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 8 horas
<i>Fecha Inicio:</i> 28 de Marzo de 2013	<i>Fecha Fin:</i> 28 de Marzo de 2013
<i>Miembro Responsable:</i> Gabriel Armendáriz	
<i>Descripción:</i> Cuando el tren tenga más de un vagón no podrá girar más de 90 grados ya que se perderá	

Tarjeta de Tarea	
Número de Tarea: T05	Historia de Usuario (Nro. Y Nombre): HU10-Programador
Nombre Tarea: El tren pueda moverse solo por los rieles o caminos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8 horas
Fecha Inicio: 29 de Marzo de 2013	Fecha Fin: 29 de Marzo de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: El tren debe moverse solo por los rieles ya que si no lo hace perderá el juego	

Tarjeta de Tarea	
Número de Tarea: T06	Historia de Usuario (Nro. Y Nombre): HU27-Sonidista
Nombre Tarea: Sonido cuando el tren este en movimiento	
Tipo de Tarea: Sonido	Puntos Estimados: 8 horas
Fecha Inicio: 29 de Marzo de 2013	Fecha Fin: 29 de Marzo de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Se debe escuchar un sonido cuando el tren ese en movimiento para que el juego tenga un aspecto real.	

Tarjetas de Tareas Segunda Iteración

Tarjeta de Tarea	
Número de Tarea: T08	Historia de Usuario (Nro. Y Nombre): HU14-Programador
Nombre Tarea: Visualizar puntaje al coger los objetos y responder las preguntas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 16 horas
Fecha Inicio: 02 de Abril de 2013	Fecha Fin: 03 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Cuando el tren coja los objetos que van apareciendo por los rieles se debe visualizar la puntuación.	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T09	<i>Historia de Usuario (Nro. Y Nombre):</i> HU15-Programador
<i>Nombre Tarea:</i> Pausar el juego al pulsar un botón.	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 8 horas
<i>Fecha Inicio:</i> 05 de Abril de 2013	<i>Fecha Fin:</i> 05 de Abril de 2013
<i>Miembro Responsable:</i> Gabriel Armendáriz	
<i>Descripción:</i> Para poder pausar el juego se debe tener un botón que al ser pulsado mande a un estado pausado.	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T10	<i>Historia de Usuario (Nro. Y Nombre):</i> HU19-Programador
<i>Nombre Tarea:</i> Visualizar el nivel en el estado preparado	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 8 horas
<i>Fecha Inicio:</i> 04 de Abril de 2013	<i>Fecha Fin:</i> 04 de Abril de 2013
<i>Miembro Responsable:</i> Milton Saltos	
<i>Descripción:</i> Para poder jugar el nivel se debe primero presentar un estado preparado en la pantalla	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T11	<i>Historia de Usuario (Nro. Y Nombre):</i> HU20-Programador
<i>Nombre Tarea:</i> Pulsar en cualquier parte de la pantalla para empezar a jugar	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 4 horas
<i>Fecha Inicio:</i> 05 de Abril de 2013	<i>Fecha Fin:</i> 05 de Abril de 2013
<i>Miembro Responsable:</i> Gabriel Armendáriz	
<i>Descripción:</i> Una vez que se quiera jugar un nivel y aparece la pantalla de preparado se debe pulsar en cualquier parte de la pantalla y se podrá iniciar a jugar	

Tarjeta de Tarea	
Número de Tarea: T12	Historia de Usuario (Nro. Y Nombre): HU28- Diseñador
Nombre Tarea: Aspecto del tren	
Tipo de Tarea: Diseño	Puntos Estimados: 8 horas
Fecha Inicio: 05 de Abril de 2013	Fecha Fin: 05 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: El tren tenga un aspecto real para hacer al juego más divertido	

Tarjeta de Tarea	
Número de Tarea: T13	Historia de Usuario (Nro. Y Nombre): HU30- Diseñador
Nombre Tarea: Los objetos tengan un aspecto de cofres, hongos y monedas	
Tipo de Tarea: Diseño	Puntos Estimados: 4 horas
Fecha Inicio: 05 de Abril de 2013	Fecha Fin: 05 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Los objetos que van apareciendo en los rieles del tren deben tener un aspecto de monedas, hongos y cofres.	

Tarjetas de Tareas Tercera Iteración

Tarjeta de Tarea	
Número de Tarea: T15	Historia de Usuario (Nro. Y Nombre): HU05- Programador
Nombre Tarea: Visualizar los créditos del juego	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8 horas
Fecha Inicio: 09 de Abril de 2013	Fecha Fin: 09 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Visualizar créditos del juego para saber quién lo realiza.	

Tarjeta de Tarea	
Número de Tarea: T16	Historia de Usuario (Nro. Y Nombre): HU24-Programador
Nombre Tarea: Salir de la aplicación (videojuego)	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4 horas
Fecha Inicio: 10 de Abril de 2013	Fecha Fin: 10 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Cuando en el menú principal se escoge la opción salir se muestre una pantalla de confirmación que permita cerrar la aplicación (videojuego).	

Tarjeta de Tarea	
Número de Tarea: T17	Historia de Usuario (Nro. Y Nombre): HU26-Programador
Nombre Tarea: Botón de sonido en la pantalla del menú principal	
Tipo de Tarea: Desarrollo	Puntos Estimados: 16 horas
Fecha Inicio: 10 de Abril de 2013	Fecha Fin: 10 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Existirá un botón de sonido en la pantalla de menú principal, para poder habilitar y deshabilitar el sonido.	

Tarjeta de Tarea	
Número de Tarea: T18	Historia de Usuario (Nro. Y Nombre): HU29-Diseñador
Nombre Tarea: Los vagones tengan un aspecto de un cuadrado	
Tipo de Tarea: Diseño	Puntos Estimados: 4 horas
Fecha Inicio: 10 de Abril de 2013	Fecha Fin: 10 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Los vagones tengan un aspecto de un cuadrado de color rojo, de modo que se muestre realista.	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T19	<i>Historia de Usuario (Nro. Y Nombre):</i> HU33-Sonidista
<i>Nombre Tarea:</i> Exista audio de fondo en las pantallas del videojuego	
<i>Tipo de Tarea:</i> Sonido	<i>Puntos Estimados:</i> 8 horas
<i>Fecha Inicio:</i> 11 de Abril de 2013	<i>Fecha Fin:</i> 11 de Abril de 2013
<i>Miembro Responsable:</i> Milton Saltos	
<i>Descripción:</i> Debe existir audio de fondo en las diferentes pantallas del videojuego, de modo que sea divertido.	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T20	<i>Historia de Usuario (Nro. Y Nombre):</i> HU04-Programador
<i>Nombre Tarea:</i> Visualizar las instrucciones del juego	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 8 horas
<i>Fecha Inicio:</i> 12 de Abril de 2013	<i>Fecha Fin:</i> 12 de Abril de 2013
<i>Miembro Responsable:</i> Gabriel Armendáriz	
<i>Descripción:</i> Visualizar las instrucciones del juego para saber cómo jugar.	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T21	<i>Historia de Usuario (Nro. Y Nombre):</i> HU11-Programador
<i>Nombre Tarea:</i> Escuchar sonido al coger los objetos con el tren	
<i>Tipo de Tarea:</i> Desarrollo	<i>Puntos Estimados:</i> 8 horas
<i>Fecha Inicio:</i> 12 de Abril de 2013	<i>Fecha Fin:</i> 12 de Abril de 2013
<i>Miembro Responsable:</i> Milton Saltos	
<i>Descripción:</i> Cuando el tren al coger los objetos se escuche un sonido de "incremento", para poder dar realismo al juego.	

Tarjetas de Tareas Cuarta Iteración

Tarjeta de Tarea	
Número de Tarea: T23	Historia de Usuario (Nro. Y Nombre): HU18-Programador
Nombre Tarea: Aparecer preguntas cada 6 objetos recogidos por el tren	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8 horas
Fecha Inicio: 16 de Abril de 2013	Fecha Fin: 16 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Las preguntas deben ir apareciendo a medida que el tren vaya cogiendo 6 objetos, para aumentar la diversión del juego.	

Tarjeta de Tarea	
Número de Tarea: T24	Historia de Usuario (Nro. Y Nombre): HU21-Programador
Nombre Tarea: Visualizar la pantalla "Fin del juego"	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8 horas
Fecha Inicio: 17 de Abril de 2013	Fecha Fin: 17 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Cuando el tren se salga de la vía o se choque con sus vagones se visualizar una pantalla de "Fin del Juego" conjuntamente con dos botones de "Nuevo Juego" y "Salir", para saber que perdí y poder intentarlo de nuevo o salir del juego.	

Tarjeta de Tarea	
Número de Tarea: T25	Historia de Usuario (Nro. Y Nombre): HU22-Programador
Nombre Tarea: Opciones de respuesta a las preguntas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8 horas
Fecha Inicio: 18 de Abril de 2013	Fecha Fin: 18 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Se debe responder las preguntas que se van mostrando mediante dos botones de afirmación (SI) o negación (NO), para poder saber el grado de conocimiento que tengo sobre la Provincia de Chimborazo y adquirir nuevo conocimiento.	

Tarjeta de Tarea	
Número de Tarea: T26	Historia de Usuario (Nro. Y Nombre): HU35- Diseñador
Nombre Tarea: Botones de navegabilidad en la pantalla instrucciones	
Tipo de Tarea: Diseño	Puntos Estimados: 8 horas
Fecha Inicio: 18 de Abril de 2013	Fecha Fin: 18 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Debe existir en la pantalla de Instrucciones botones de navegabilidad entre las pantallas de esta opción.	

Tarjeta de Tarea	
Número de Tarea: T27	Historia de Usuario (Nro. Y Nombre): HU32- Diseñador
Nombre Tarea: Botón de regreso al menú principal	
Tipo de Tarea: Diseño	Puntos Estimados: 8 horas
Fecha Inicio: 19 de Abril de 2013	Fecha Fin: 19 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Tiene que existir en la pantalla de créditos, instrucciones, puntuaciones, niveles y salir un botón de regreso al menú principal.	

Tarjeta de Tarea	
Número de Tarea: T28	Historia de Usuario (Nro. Y Nombre): HU25- Programador
Nombre Tarea: Acceder a las redes sociales	
Tipo de Tarea: Diseño	Puntos Estimados: 8 horas
Fecha Inicio: 19 de Abril de 2013	Fecha Fin: 19 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Debe existir 2 botones para poder acceder a las redes sociales como Facebook o Twitter, para conocer opiniones del juego e información adicional del mismo	

Tarjetas de Tareas Quinta Iteración

Tarjeta de Tarea	
Número de Tarea: T30	Historia de Usuario (Nro. Y Nombre): HU31- Diseñador
Nombre Tarea: Exista animaciones en las pantallas	
Tipo de Tarea: Diseño	Puntos Estimados: 16 horas
Fecha Inicio: 23 de Abril de 2013	Fecha Fin: 24 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Debe existir animaciones en la pantalla del menú principal, créditos, instrucciones, niveles y salir del videojuego.	

Tarjeta de Tarea	
Número de Tarea: T31	Historia de Usuario (Nro. Y Nombre): HU13- Programador
Nombre Tarea: Ingresar nombre, apellido y foto(opcional)	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8 horas
Fecha Inicio: 25 de Abril de 2013	Fecha Fin: 25 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Se debe ingresar los datos personales como nombre y apellido y foto (opcional), para poder visualizar junto a la puntuación alcanzada.	

Tarjeta de Tarea	
Número de Tarea: T32	Historia de Usuario (Nro. Y Nombre): HU16- Programador
Nombre Tarea: Cancelar ingreso de datos del jugador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4 horas
Fecha Inicio: 25 de Abril de 2013	Fecha Fin: 25 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Se debe cancelar el ingreso de los datos en caso de que este así lo decida.	

Tarjeta de Tarea	
Número de Tarea: T33	Historia de Usuario (Nro. Y Nombre): HU23-Programador
Nombre Tarea: Cada nivel tenga 5 preguntas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4 horas
Fecha Inicio: 25 de Abril de 2013	Fecha Fin: 25 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: En cada nivel debe haber un máximo de 5 preguntas, para que el juego no sea aburrido	

Tarjeta de Tarea	
Número de Tarea: T34	Historia de Usuario (Nro. Y Nombre): HU17-Programador
Nombre Tarea: Aumentar vagones al tren	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4 puntos
Fecha Inicio: 26 de Abril de 2013	Fecha Fin: 26 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: El tren debe aumentar de vagones a medida que vaya cogiendo 3 objetos de los rieles.	

Tarjeta de Tarea	
Número de Tarea: T35	Historia de Usuario (Nro. Y Nombre): HU34-Diseñador
Nombre Tarea: Mostrar varios campos en la pantalla de puntuaciones.	
Tipo de Tarea: Diseño	Puntos Estimados: 4 horas
Fecha Inicio: 26 de Abril de 2013	Fecha Fin: 26 de Abril de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: En la pantalla de puntuaciones se debe mostrar el nombre, apellido, puntuación, número de preguntas correctas e incorrectas contestadas de los 10 jugadores con más altas puntuaciones, de modo que se pueda apreciar mejor los datos de los jugadores.	

Tarjeta de Tarea	
Número de Tarea: T36	Historia de Usuario (Nro. Y Nombre): HU37-Programador
Nombre Tarea: Contenido de preguntas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8 horas
Fecha Inicio: 26 de Abril de 2013	Fecha Fin: 26 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: El contenido de las preguntas debe ser sobre la provincia de Chimborazo, de modo que pueda complementar el conocimiento del jugador.	

Tarjetas de Tareas Sexta Iteración

Tarjeta de Tarea	
Número de Tarea: T38	Historia de Usuario (Nro. Y Nombre): HU39-Diseñador
Nombre Tarea: Responder preguntas incorrectamente	
Tipo de Tarea: Diseño	Puntos Estimados: 8 horas
Fecha Inicio: 30 de Abril de 2013	Fecha Fin: 30 de Abril de 2013
Miembro Responsable: Milton Saltos	
Descripción: Cuando se responda una pregunta incorrectamente de debe mostrar un mensaje de "INCORRECTO" con una X de color rojo y un botón de "continuar" para seguir jugando, conjuntamente se debe escuchar un sonido de incorrecto, de modo que pueda saber que se respondió incorrectamente.	

Tarjeta de Tarea	
Número de Tarea: T39	Historia de Usuario (Nro. Y Nombre): HU40-Diseñador
Nombre Tarea: Responder pregunta incorrectamente	
Tipo de Tarea: Diseño	Puntos Estimados: 8 horas
Fecha Inicio: 01 de Mayo de 2013	Fecha Fin: 01 de Mayo de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Al responder una pregunta incorrectamente se tiene que mostrar la respuesta correcta, de modo pueda saber que es lo correcto.	

Tarjeta de Tarea	
Número de Tarea: T40	Historia de Usuario (Nro. Y Nombre): HU41-Programador
Nombre Tarea: La puntuación del juego se vaya acumulando	
Tipo de Tarea: Programación	Puntos Estimados: 4 horas
Fecha Inicio: 01 de Mayo de 2013	Fecha Fin: 01 de Mayo de 2013
Miembro Responsable: Milton Saltos	
Descripción: La puntuación del juego de debe ir acumulando nivel tras nivel completado, para conocer el total alcanzado al finalizar el juego.	

Tarjeta de Tarea	
Número de Tarea: T41	Historia de Usuario (Nro. Y Nombre): HU42-Programador
Nombre Tarea: Disminución de la puntuación	
Tipo de Tarea: Programación	Puntos Estimados: 4 horas
Fecha Inicio: 01 de Mayo de 2013	Fecha Fin: 01 de Mayo de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: La puntuación debe disminuir cuando se responda incorrectamente en 10 puntos, de modo que no pueda alcanzar la máxima puntuación	

Tarjeta de Tarea	
Número de Tarea: T42	Historia de Usuario (Nro. Y Nombre): HU43-Jugador
Nombre Tarea: Completar todos los niveles	
Tipo de Tarea: Programación	Puntos Estimados: 4 horas
Fecha Inicio: 02 de Mayo de 2013	Fecha Fin: 02 de Mayo de 2013
Miembro Responsable: Gabriel Armendáriz	
Descripción: Cuando se complete todos los niveles y se responda correctamente todas las preguntas se mostrará una pantalla de felicitaciones, de modo que pueda sentirse elogiado	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T43	<i>Historia de Usuario (Nro. Y Nombre):</i> HU36-Sonidista
<i>Nombre Tarea:</i> Sonido de fondo al iniciar el Splash	
<i>Tipo de Tarea:</i> Sonido	<i>Puntos Estimados:</i> 2 horas
<i>Fecha Inicio:</i> 02 de Mayo de 2013	<i>Fecha Fin:</i> 02 de Mayo de 2013
<i>Miembro Responsable:</i> Milton Saltos	
<i>Descripción:</i> Se escuche un sonido al iniciar el juego en el Splash que aparece, de modo que sea divertido.	

Tarjeta de Tarea	
<i>Número de Tarea:</i> T44	<i>Historia de Usuario (Nro. Y Nombre):</i> HU44-Jugador
<i>Nombre Tarea:</i> Completar un nivel	
<i>Tipo de Tarea:</i> Programación	<i>Puntos Estimados:</i> 2 horas
<i>Fecha Inicio:</i> 02 de Mayo de 2013	<i>Fecha Fin:</i> 02 de Mayo de 2013
<i>Miembro Responsable:</i> Milton Saltos	
<i>Descripción:</i> Cuando se complete un nivel correctamente se muestre una pantalla de nivel completado, con un botón de "CONTINUAR" y otro de "SALIR", de modo que pueda saber que complete el nivel correctamente.	

ANEXO III

**CÓDIGO DE LAS CLASES PRINCIPALES DEL
VIDEOJUEGO.**

Paquete com.metodologia.videojuegodav

Clase Entrada

```
package com.metodologia.videojuegodav;
import java.util.List;
public interface Entrada {
    public static class KeyEvent {
        public static final int KEY_DOWN = 0;
        public static final int KEY_UP = 1;
        public int type;
        public int keyCode;
        public char keyChar;
        /*
         * (non-Javadoc)
         * @see java.lang.Object#toString()
         */
        public String toString(){
            StringBuilder builder = new StringBuilder();
            if(type == KEY_DOWN)
                builder.append("tecla pulsada, ");
            else
                builder.append("tecla levantada, ");
            builder.append(keyCode);
            builder.append(" , ");
            builder.append(keyChar);
            return builder.toString();
        }
    }
}
/*
 * Clase para touch event
 */
public static class TouchEvent{
    public static final int TOUCH_DDOWN=0;
    public static final int TOUCH_UP=1;
    public static final int TOUCH_DRAGGED=2;

    public int type;
    public int x , y;
    public int pointer;
    public String toString(){
        StringBuilder builder = new StringBuilder();
        if(type == TOUCH_DDOWN)
            builder.append("touch down, ");
        else if(type == TOUCH_DRAGGED)
            builder.append("touch dragged, ");
        else
            builder.append("touch up, ");
        builder.append(pointer);
        builder.append(" , ");
        builder.append(x);
        builder.append(" , ");
        builder.append(y);
    }
}
```

```

        return builder.toString();
    }
}

//devuelve si la tecla esta pulsada o no
public boolean isKeyPressed(int KeyCode);
//metodos para los toques de pantalla
public boolean isTouchDown(int pointer);
public int getTouchX(int pointer);
public int getTouchY(int pointer);
//metodos para acelerometro
public float getAccelX();
public float getAccelY();
public float getAccelZ();
// metodos usados para el manejo basados en eventos
public List<KeyEvent> getKeyEvent();
public List<TouchEvent> getTouchEvent();
}

```

Clase ArchivoES

```

package com.metodologia.videojuegodav;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

public interface ArchivoES {
    public InputStream leerAsset(String nombreArchivo) throws IOException;
    public InputStream leerArchivo(String nombreArchivo) throws IOException;
    public OutputStream escribirArchivo(String nombreArchivo) throws IOException;
}

```

Clase Juego

```

package com.metodologia.videojuegodav;

public interface Juego {
    public Entrada getEntrada();
    public ArchivoES getArchivoES();
    public Graficos getGraficos();
    public Audio getAudio();
    public void setScreen(Pantalla pantalla);
    public Pantalla getCurrentScreen();
    public Pantalla getStartScreen();
}

```

Clase Jugador

```

package com.metodologia.videojuegodav

```

```

import java.io.Serializable;
public class Persona implements Serializable{
    // Atributos de la clase
    private int id;
    private String nombre;
    private String apellido;
    private int puntuacion;
    private int preguntascorrectas;
    private int preguntasincorrectas;
    private String ruta_imagen;
    /**
     * Constructor por defecto.
     */
    public Jugador(){
    }

    public Jugador(String p_nombre, String p_apellido, int p_puntuacion, int p_pregcorrectas, int
    p_pregincorrectas, String p_ruta_imagen){
        setNombre(p_nombre);
        setApellido(p_apellido);
        setPuntuacion(p_puntuacion);
        setPregcorrectas(p_pregcorrectas);
        setPregincorrectas(p_pregincorrectas);
        setRutaImagen(p_ruta_imagen);
    }
    /**
     * Constructor que se le pasa 7 parametros.
     * @param p_id
     * @param p_nombre
     * @param p_apellido
     * @param p_puntuacion
     * @param p_preguntascorrectas
     * @param p_preguntasincorrectas
     * @param p_ruta_imagen
     */
    public Persona(int p_id, String p_nombre, String p_apellido, int p_puntuacion, int
    p_pregcorrectas, int p_pregincorrectas, String p_ruta_imagen){
        setId(p_id);
        setNombre(p_nombre);
        setApellido(p_apellido);
        setPuntuacion(p_puntuacion);
        setPregcorrectas(p_pregcorrectas);
        setPregincorrectas(p_pregincorrectas);
        setRutaImagen(p_ruta_imagen);
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

```

```
}

public String getApellido() {
    return this.apellido;
}

public void setApellido(String apellido) {
    this.apellido = apellido;
}

public int getPuntuacion() {
    return puntuacion;
}

public void setPuntuacion(int puntuacion) {
    this.puntuacion= puntuacion;
}

public void setPregcorrectas(int pregcorrectas) {
    this.preguntascorrectas= pregcorrectas;
}

public int getPregcorrectas() {
    return preguntascorrectas;
}

public void setPregincorrectas(int pregincorrectas) {
    this.preguntasincorrectas= pregincorrectas;
}

public int getPregincorrectas() {
    return preguntasincorrectas;
}

public void setId(int id){
    this.id = id;
}

public int getId(){
    return this.id;
}

public String getRutaImagen(){
    return this.ruta_imagen;
}

public void setRutaImagen(String ruta_imagen){
    // Si ruta_imagen es null entonces se pone un valor predeterminado.
    if(ruta_imagen == null){
        this.ruta_imagen = "No tiene imagen.";
    }else{
```

```

        this.ruta_imagen = ruta_imagen;
    }
} // Fin de la clase Juegador.

```

Paquete com.metodologia.videojuegodav.implementacion

Clase VideojuegoSonido

```

package com.metodologia.videojuegodav.implementacion;
import android.media.SoundPool;
import com.metodologiaa.videojuegodav.Sonido;

public class VideojuegoSonido implements Sonido {
    int soundId;
    SoundPool soundPool;
    public VideojuegoSonido(SoundPool soundPool, int soundId){
        this.soundId = soundId;
        this.soundPool = soundPool;
    }

    public void play(float volume) {
        soundPool.play(soundId, volume, volume, 0, 0, 1);
    }

    public void dispose() {
        soundPool.unload(soundId);
    }
}

```

Clase VideojuegoMusica

```

package com.metodologia.videojuegodav.implementacion;
import java.io.IOException;
import android.media.MediaPlayer.OnCompletionListener;
import android.media.MediaPlayer;
import android.content.res.AssetFileDescriptor;
import com.metodologiaa.videojuegodav.Musica;

public class VideojuegoMusica implements Musica, OnCompletionListener {
    MediaPlayer mediaPlayer;
    boolean isPrepared = false;

    public VideojuegoMusica(AssetFileDescriptor assetFileDescriptor) {
        mediaPlayer = new MediaPlayer();
        try {
            mediaPlayer.setDataSource(assetFileDescriptor.getFileDescriptor(),
                assetFileDescriptor.getStartOffset(),
                assetFileDescriptor.getLength());
        }
    }
}

```

```

        mediaPlayer.prepare();
        isPrepared = true;
        mediaPlayer.setOnCompleteListener(this);
    } catch (Exception e) {
        throw new RuntimeException("No se ha podido cargar la musica");
    }
}

public void dispose() {
    if(mediaPlayer.isPlaying())
        mediaPlayer.stop();
    mediaPlayer.release();
}

public boolean isLooping() {
    return mediaPlayer.isLooping();
}

public boolean isPlaying() {
    return mediaPlayer.isPlaying();
}

public void start() {
    mediaPlayer.start();
}

public boolean isStopped() {
    return !isPrepared;
}

public void pause() {
    if(mediaPlayer.isPlaying())
        mediaPlayer.pause();
}

public void play() {
    if(mediaPlayer.isPlaying())
        return;
    try {
        synchronized (this) {
            if(!isPrepared)
                mediaPlayer.prepare();
            mediaPlayer.start();
        }
    } catch (IllegalStateException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

```

public void setLooping(boolean looping) {
    mediaPlayer.setLooping(isLooping());
}

public void setVolume(float volume) {
    mediaPlayer.setVolume(volume, volume);
}

public void stop() {
    mediaPlayer.stop();
    synchronized (this) {
        isPrepared= false;
    }
}

public void onCompletion(MediaPlayer arg0) {
    synchronized (this) {
        isPrepared = false;
    }
}
}

```

Clase VideojuegoAudio

```

package com.metodologia.videojuegodav.implementacion;
import java.io.IOException;
import android.app.Activity;
import android.content.res.AssetFileDescriptor;
import android.content.res.AssetManager;
import android.media.AudioManager;
import android.media.SoundPool;
import com.metodologiaa.videojuegodav.Audio;
import com.metodologiaa.videojuegodav.Musica;
import com.metodologiaa.videojuegodav.Sonido;

public class VideojuegoAudio implements Audio {
    AssetManager assets;
    SoundPool soundPool;
    /*
     * constructor VideojuegoAudio
     */
    public VideojuegoAudio(Activity activity){
        activity.setVolumeControlStream(AudioManager.STREAM_MUSIC);
        this.assets = activity.getAssets();
        this.soundPool = new SoundPool(20, AudioManager.STREAM_MUSIC, 0);
    }

    public Musica nuevaMusica(String nombreArchivo) {
        try {
            AssetFileDescriptor assetFileDescriptor = assets.openFd(nombreArchivo);
            return new VideojuegoMusica(assetFileDescriptor);
        } catch (IOException e) {

```

```

        throw new RuntimeException("no se ha podido cargar el archivo" +
            nombreArchivo + "");
    }
}

public Sonido nuevoSonido(String filename) {
    try {
        AssetFileDescriptor assetFileDescriptor = assets.openFd(filename);
        int soundId = soundPool.load(assetFileDescriptor, 0);
        return new VideojuegoSonido(soundPool, soundId);
    } catch (IOException e) {
        throw new RuntimeException("no se ha podido cargar el archivo" + filename
            + "");
    }
}
}

```

Clase VideojuegoGraficos

```

package com.metodologia.videojuegodav.implementacion;
import java.io.IOException;
import java.io.InputStream;
import android.content.res.AssetManager;
import android.graphics.Bitmap;
import android.graphics.Bitmap.Config;
import android.graphics.BitmapFactory;
import android.graphics.BitmapFactory.Options;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.Paint.Align;
import android.graphics.Paint.Style;
import android.graphics.Rect;
import com.metodologiaa.videojuegodav.Graficos;
import com.metodologiaa.videojuegodav.Pixmap;

```

```

public class VideojuegoGraficos implements Graficos {
    // cargar instancias bitmao
    AssetManager assets;
    // represena el frame buffer artificial
    Bitmap framebuffer;
    // Dibujar el frame buffer
    Canvas canvas;
    // Dibuar
    Paint paint;
    // implementar los metodos de drawBitmap
    Rect srcRect = new Rect();
    Rect dstRect = new Rect();

    public VideojuegoGraficos(AssetManager assets, Bitmap framebuffer) {
        this.assets = assets;
        this.frameBuffer = framebuffer;
    }
}

```

```

        this.canvas = new Canvas(frameBuffer);
        this.paint = new Paint();
    }

    public Pixmap newPixmap(String fileName, PixmapFormat format) {
        Config config = null;
        if (format == PixmapFormat.RGB565)
            config = Config.RGB_565;
        else if (format == PixmapFormat.ARGB4444)
            config = Config.ARGB_4444;
        else
            config = Config.ARGB_8888;
        // configuramos el formato de color preferido
        Options options = new Options();
        options.inPreferredConfig = config;
        InputStream in = null;
        Bitmap bitmap = null;
        try {
            in = assets.open(fileName);
            bitmap = BitmapFactory.decodeStream(in);
            if (bitmap == null)
                throw new RuntimeException(
                    "No se ha podido cargar bitmap desde asset" +
                        filename + "");
        } catch (IOException e) {
            throw new RuntimeException(
                "No se ha podido cargar bitmap desde asset" + fileName
                    + "");
        } finally {
            if (in != null) {
                try {
                    in.close();
                } catch (IOException e) {
                }
            }
        }

        if (bitmap.getConfig() == Config.RGB_565)
            format = PixmapFormat.RGB565;
        else if (bitmap.getConfig() == Config.ARGB_4444)
            format = PixmapFormat.ARGB4444;
        else
            format = PixmapFormat.ARGB8888;
        return new VideojuegoPixmap(bitmap, format);
    }

    public void clear(int color) {
        canvas.drawRGB((color & 0xff0000) >> 16, (color & 0xff00) >> 8,
            (color & 0xff));
    }

    public void drawPixel(int x, int y, int color) {

```

```

        paint.setColor(color);
        canvas.drawPoint(x, y, paint);
    }

    public void drawLine(int x, int y, int x2, int y2, int color) {
        paint.setColor(color);
        canvas.drawLine(x, y, x2, y2, paint);
    }

    public void drawText (String text, float x, float y)
    {
        paint.setAntiAlias(true);
        paint.setARGB(255, 255, 255, 255);
        paint.setTextSize(17);
        paint.setStyle(Style.FILL);
        paint.setFakeBoldText(true);
        paint.setTextAlign(Align.CENTER);
        canvas.drawText(text, x, y, paint);
    }

    public void drawRect(int x, int y, int width, int height, int color) {
        paint.setColor(color);
        paint.setStyle(Style.FILL);
        canvas.drawRect(x, y, x + width - 1, y + height - 1, paint);
    }

    public void drawPixmap(Pixmap pixmap, int x, int y, int srcX, int srcY,
        int srcWidth, int srcHeight) {
        srcRect.left = srcX;
        srcRect.top = srcY;
        srcRect.right = srcX + srcWidth - 1;
        srcRect.bottom = srcY + srcHeight - 1;
        dstRect.left = x;
        dstRect.top = y;
        dstRect.right = x + srcWidth - 1;
        dstRect.bottom = y + srcHeight - 1;
        canvas.drawBitmap(((VideojuegoPixmap) pixmap).bitmap, srcRect, dstRect, null);
    }

    public void drawPixmap(Pixmap pixmap, int x, int y) {
        canvas.drawBitmap(((VideojuegoPixmap) pixmap).bitmap, x, y, null);
    }

    public int getWidth() {
        return framebuffer.getWidth();
    }

    public int getHeight() {
        return framebuffer.getHeight();
    }
}

```

Clase VideojuegoFastRenderView

```
package com.metodologia.videojuegodav.implementacion;
import com.metodologiaa.videojuegodav.juego.MainMenuScreen;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Rect;
import android.view.SurfaceHolder;
import android.view.SurfaceView;

public class VideojuegoFastRenderView extends SurfaceView implements Runnable {
    VideojuegoJuego juego;
    Bitmap frameBuffer;
    Thread renderThread = null;
    SurfaceHolder holder;
    MainMenuScreen view;
    volatile boolean running = false;
    static final long FPS = 10;

    public VideojuegoFastRenderView(VideojuegoJuego juego, Bitmap frameBuffer) {
        super(juego);
        this.juego = juego;
        this.frameBuffer = frameBuffer;
        this.holder = getHolder();
    }

    public void resume() {
        running = true;
        renderThread = new Thread(this);
        renderThread.start();
    }

    public void run() {
        Rect dstRect = new Rect();
        long startTime = System.nanoTime();
        long sleepTime;
        long ticksPS = 1000000 / FPS;
        while (running) {
            if (!holder.getSurface().isValid())
                continue;
            float deltaTime = (System.nanoTime() - startTime) / 1000000000.0f;
            startTime = System.nanoTime();
            Canvas canvas = holder.lockCanvas();
            juego.getCurrentScreen().update(deltaTime);
            juego.getCurrentScreen().present(deltaTime);
            canvas.getClipBounds(dstRect);
            canvas.drawBitmap(frameBuffer, null, dstRect, null);
            holder.unlockCanvasAndPost(canvas);
            try {

                juego.getCurrentScreen().onDraw1();
            }
            finally {
```

```

        if (canvas != null) {
            }
        }
        sleepTime = ticksPS - System.currentTimeMillis() - startTime;
        try {
            if (sleepTime > 0)
                Thread.sleep(sleepTime);
            else
                Thread.sleep(50);
        } catch (Exception e) {
        }
    }
}

public void pause() {
    running = false;
    while(true){
        try {
            renderThread.join();
            break;
        } catch (InterruptedException e) {
            // retry
        }
    }
}
}
}

```

Clase VideojuegoEntrada

package com.metodologia.videojuegodav.implementacion;

import java.util.List;

import android.content.Context;

import android.os.Build.VERSION;

import android.view.View;

import com.metodologiaa.videojuegodav.Entrada;

public class VideojuegoEntrada **implements** Entrada{

ManejadorAcelerometro **manejadorAccel**;

ManejadorKeyboard **manejadorKey**;

ManejadorTouch **manejadorTouch**;

public VideojuegoEntrada(Context context, View view, **float** scaleX, **float** scaleY){

manejadorAccel = **new** ManejadorAcelerometro(context);

manejadorKey = **new** ManejadorKeyboard(view);

if(Integer.parseInt(VERSION.SDK) < 3)

manejadorTouch = **new** ManejadorMultiTouch(view, scaleX, scaleY);

else

manejadorTouch = **new** ManejadorSingleTouch(view, scaleX, scaleY);

}

public boolean isKeyPressed(**int** KeyCode) {

return **manejadorKey**.isKeyPressed(KeyCode);

```

    }

    public boolean isTouchDown(int pointer) {
        return manejadorTouch.isTouchDown(pointer);
    }

    public int getTouchX(int pointer) {
        return manejadorTouch.getTouchX(pointer);
    }

    public int getTouchY(int pointer) {
        return manejadorTouch.getTouchY(pointer);
    }

    public float getAccelX() {
        return manejadorAccel.getAccelX();
    }

    public float getAccelY() {
        return manejadorAccel.getAccelY();
    }

    public float getAccelZ() {
        return manejadorAccel.getAccelZ();
    }

    public List<KeyEvent> getKeyEvent() {
        return manejadorKey.getKeyEvents();
    }

    public List<TouchEvent> getTouchEvents() {
        return manejadorTouch.getTouchEvents();
    }
}

```

Clase ManejadorSingleTouch

```

package com.metodologiaa.videojuegodav.implementacion;
import java.util.ArrayList;
import java.util.List;
import android.view.MotionEvent;
import android.view.View;
import com.metodologiaa.videojuegodav.Agrupamiento;
import com.metodologiaa.videojuegodav.Agrupamiento.PoolObjectFactory;
import com.metodologiaa.videojuegodav.Entrada.TouchEvent;

public class ManejadorSingleTouch implements ManejadorTouch {
    boolean isTpuched;
    int touchX;
    int touchY;
    Agrupamiento<TouchEvent> touchEventAgrup;
}

```

```

List<TouchEvent> touchEvents = new ArrayList<TouchEvent>();
List<TouchEvent> touchEventsBuffer = new ArrayList<TouchEvent>();
float scaleX;
float scaleY;
public ManejadorSingleTouch(View view, float scaleX, float scaleY) {
    PoolObjectFactory<TouchEvent> factory = new PoolObjectFactory<TouchEvent>() {
        public TouchEvent createObject() {
            return new TouchEvent();
        }
    };
    touchEventAgrup = new Agrupamiento<TouchEvent>(factory, 100);
    view.setOnTouchListener(this);

    this.scaleX = scaleX;
    this.scaleY = scaleY;
}

```

```

public boolean onTouch(View v, MotionEvent event) {
    synchronized (this) {
        TouchEvent touchEvent = touchEventAgrup.newObject();
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                touchEvent.type = TouchEvent.TOUCH_DOOWN;
                isTpuched = true;
                break;
            case MotionEvent.ACTION_MOVE:
                touchEvent.type = TouchEvent.TOUCH_DRAGGED;
                isTpuched = true;
                break;
            case MotionEvent.ACTION_CANCEL:
            case MotionEvent.ACTION_UP:
                touchEvent.type = TouchEvent.TOUCH_UP;
                isTpuched = false;
                break;
        }
        touchEvent.x = touchX = (int) (event.getX() * scaleX);
        touchEvent.y = touchY = (int) (event.getY() * scaleY);
        touchEventsBuffer.add(touchEvent);
        return true;
    }
}

```

```

public boolean isTouchDown(int pointer) {
    synchronized (this) {
        if (pointer == 0)
            return isTpuched;
        else
            return false;
    }
}

```

```

public int getTouchX(int pointer) {

```

```

        synchronized (this) {
            return touchX;
        }
    }

    public int getTouchY(int pointer) {
        synchronized (this) {
            return touchY;
        }
    }

    public List<TouchEvent> getTouchEvents() {
        synchronized (this) {
            int len = touchEvents.size();
            for (int i = 0; i < len; i++)
                touchEventAgrup.free(touchEvents.get(i));
            touchEvents.clear();
            touchEvents.addAll(touchEventsBuffer);
            touchEventsBuffer.clear();
            return touchEvents;
        }
    }
}

```

Clase VideojuegoJuego

```

package com.metodologia.videojuegodav.implementacion;
import android.app.Activity;
import android.content.Context;
import android.content.res.Configuration;
import android.graphics.Bitmap;
import android.graphics.Bitmap.Config;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.PowerManager;
import android.os.PowerManager.WakeLock;
import android.view.Window;
import android.view.WindowManager;
import com.metodologiaa.videojuegodav.Audio;
import com.metodologiaa.videojuegodav.ArchivoES;
import com.metodologiaa.videojuegodav.Juego;
import com.metodologiaa.videojuegodav.Graficos;
import com.metodologiaa.videojuegodav.Entrada;
import com.metodologiaa.videojuegodav.Pantalla;
import com.metodologiaa.videojuegodav.juego.Configuraciones;
import com.metodologiaa.videojuegodav.splash.R;

public abstract class VideojuegoJuego extends Activity implements Juego {
    VideojuegoFastRenderView renderView;
    Graficos graficos;
    Audio audio;
    Entrada entrada;
    ArchivoES archivoES;
}

```

```
Pantalla pantalla;  
WakeLock wakeLock;  
private MediaPlayer sound1;
```

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    requestWindowFeature(Window.FEATURE_NO_TITLE);  
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
        WindowManager.LayoutParams.FLAG_FULLSCREEN);  
    boolean isLandscape = getResources().getConfiguration().orientation ==  
        Configuration.ORIENTATION_LANDSCAPE;  
    int frameBufferWidth = isLandscape ? 480 : 320;  
    int frameBufferHeight = isLandscape ? 320 : 480;  
    Bitmap frameBuffer = Bitmap.createBitmap(frameBufferWidth,  
        frameBufferHeight, Config.RGB_565);  
    float scaleX = (float) frameBufferWidth /  
        getWindowManager().getDefaultDisplay().getWidth();  
    float scaleY = (float) frameBufferHeight /  
        getWindowManager().getDefaultDisplay().getHeight();  
    renderView = new VideojuegoFastRenderView(this, frameBuffer);  
    graficos = new VideojuegoGraficos(getAssets(), frameBuffer);  
    archivoES = new VideojuegoArchivoES(getAssets());  
    audio = new VideojuegoAudio(this);  
    entrada = new VideojuegoEntrada(this, renderView, scaleX, scaleY);  
    pantalla = getStartScreen();  
    setContentView(renderView);  
    PowerManager powerManager = (PowerManager)  
        getSystemService(Context.POWER_SERVICE);  
    wakeLock = powerManager.newWakeLock(PowerManager.FULL_WAKE_LOCK,  
        "GLGame");  
}
```

```
protected void onStart() {  
    super.onStart();  
}
```

```
public void onResume(){  
    super.onResume();  
    wakeLock.acquire();  
    pantalla.resume();  
    renderView.resume();  
}
```

```
public void onPause(){  
    super.onPause();  
    wakeLock.release();  
    renderView.pause();  
    pantalla.pause();  
    if (isFinishing())  
        pantalla.dispose();  
}
```

```

public Entrada getEntrada(){
    return entrada;
}

public ArchivoES getArchivoES(){
    return archivoES;
}

public Graficos getGraficos(){
    return graficos;
}

public Audio getAudio(){
    return audio;
}

public void setScreen(Pantalla pantalla){
    if(pantalla == null)
        throw new IllegalArgumentException("Pantalla no debe ser null");
    this.pantalla.pause();
    this.pantalla.dispose();
    pantalla.resume();
    pantalla.update(0);
    this.pantalla = pantalla;
}

public Pantalla getCurrentScreen() {
    return pantalla;
}

public void reproducirmusica() {
    if (sound1.isPlaying()) {
    } else {
        sound1.start();
    }
    if(sound1.isLooping())
    {
        sound1.start();
    }
}

public void reproducirmusicaparar() {

    if (sound1.isPlaying()) {
        sound1.stop();
    }
}

public void onDestroy() {
    super.onDestroy();
    finish();
}
}

```

Clase DatabaseHandler

```
package com.metodologia.videojuegodav.implementacion;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import com.metodologiaa.videojuegodav.Persona;

public class DatabaseHandler extends SQLiteOpenHelper{
    // Ruta por defecto de las bases de datos en el sistema Android.
    private static String RUTA_BASE_DATOS =
"/data/data/com.metodologiaa.videojuegodav.splash/databases/";
    // Nombre de la Base de Datos.
    private static String NOMBRE_BASE_DATOS = "Tesis";
    // Version de la Base de Datos.
    private static final int VERSION_BASE_DATOS = 1;
    // Objeto Base de Datos.
    private SQLiteDatabase base_datos;
    // Objeto Contexto.
    private Context contexto;
    // Constante privada
    private String SENTENCIA_SQL_CREAR_BASE_DATOS_PERSONAS =
"CREATE TABLE if not exists personas (_id INTEGER PRIMARY KEY
autoincrement, " + "nombre TEXT, apellido TEXT, puntuacion INTEGER,
pregcorrectas INTEGER, pregincorrectas INTEGER, ruta_imagen TEXT)";

    public DatabaseHandler(Context contexto) {
        super(contexto, NOMBRE_BASE_DATOS, null, VERSION_BASE_DATOS);
        this.contexto = contexto;
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        // Se ejecuta la sentencia SQL de creación de la tabla Jugadores.
        db.execSQL(SENTENCIA_SQL_CREAR_BASE_DATOS_PERSONAS);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Se elimina la versión anterior de la tabla Jugadores.
        db.execSQL("DROP TABLE IF EXISTS Personas");

        // Se crea la nueva versión de la tabla jugadores.
        db.execSQL(SENTENCIA_SQL_CREAR_BASE_DATOS_PERSONAS);
    }

    /**
     * Metodo publico para insertar una nueva persona.
     */
    public void insertarPersona(Persona persona){
        ContentValues valores = new ContentValues();
        valores.put("nombre", persona.getNombre());
        valores.put("apellido", persona.getApellido());
    }
}
```

```

        valores.put("puntuacion", persona.getPuntuacion());
        valores.put("pregcorrectas", persona.getPregcorrectas());
        valores.put("pregincorrectas", persona.getPregincorrectas());
        valores.put("ruta_imagen", persona.getRutaImagen());
        this.getWritableDatabase().insert("Personas", null, valores);
    }

    /**
     * Metodo publico que cierra la base de datos.
     */
    public void cerrar(){
        this.close();
    }

    /**
     * Metodo publico que devuelve todos los jugadores.
     * @return
     */
    public Cursor obtenerTodasPersonas(){
        String[] columnas = new String[]{"_id", "nombre", "apellido",
        "puntuacion", "pregcorrectas", "pregincorrectas", "ruta_imagen"};

        Cursor cursor = this.getReadableDatabase().query("Personas", columnas, null,
        null, null, null, "puntuacion DESC", Integer.toBinaryString(10));
        if(cursor != null) {
            cursor.moveToFirst();
        }
        return cursor;
    }
}

```

Clases Paquete com.metodologia.videojuegodav.juego

Clase Mundo

```

package com.metodologia.videojuegodav.juego;
import java.util.Random;

```

```

public class Mundo {
    static final int mundo_Ancho = 15;
    static final int mundo_Alto = 8;
    static final int incremento_puntuacion =10;
    static final float tick_inicial = 0.7f;
    static final float tick_decremento = 0.0000000000003f;
    public MovimientoObjeto movimientoObjeto;
    public Objeto objeto;
    public boolean finalJuego = false;
    public boolean finalJuego1 = false;
    public int puntuacion = 0;
    public int puntuacion1 = 0;

```

```

boolean campos[][] = new boolean[mundo_Ancho][mundo_Alto];
Random random = new Random();
float tiempoTick = 0;
static float tick = tick_inicial;
public Mundo() {
    movimientoObjeto = new MovimientoObjeto();
    colocarObjeto();
}

private void colocarObjeto(){
    int objetoX =8;
    int objetoY = 1;
    int a = random.nextInt(6);
    if(a==1){
        objetoX =11;
        objetoY = random.nextInt(mundo_Alto);
    }
    if(a==2){
        objetoX =3;
        objetoY =3;
    }
    if(a==3){
        objetoX =7;
        objetoY = 3;
    }
    if(a==4){
        objetoX =3;
        objetoY = 8;
    }
    if(a==0){
        objetoX =11;
        objetoY = 8;
    }
    if(a==5){
        objetoX =3;
        objetoY = 0;
    }
    if(a==6){
        objetoX = random.nextInt(mundo_Ancho);
        objetoY =7;
    }

        if(objetoX >= mundo_Ancho){
            objetoX = 0;
            objetoY += 7;
            if (objetoY >= mundo_Alto){
                objetoY = 0;
            }
        }
    objeto = new Objeto(objetoX, objetoY, random.nextInt(6));
}

```

```

public void update(float deltaTime){
    if(finalJuego)
        return
    tiempoTick += deltaTime;
    while(tiempoTick > tick){
        tiempoTick -= tick;
        movimientoObjeto.avance();
        if(Configuraciones.soundEnabled)
        {
            Assets.trenjuego.play(1);
        }
        if(movimientoObjeto.comprobarChoque() ){
            finalJuego1 = true;
            return;
        }
        Tripulacion head = movimientoObjeto.partes.get(0);
        if(head.x == objeto.x && head.y == objeto.y){
            puntuacion += incremento_puntuacion;
            puntuacion1 += incremento_puntuacion;
            if(PantallaJuego.numvagon==3 || PantallaJuego.numvagon==6 ||
            PantallaJuego.numvagon==9 || PantallaJuego.numvagon==12 ||
            PantallaJuego.numvagon==15 || PantallaJuego.numvagon==18 ||
            PantallaJuego.numvagon==21 || PantallaJuego.numvagon==24 ||
            PantallaJuego.numvagon==27 || PantallaJuego.numvagon==30)
            movimientoObjeto.abordaje();
            if(movimientoObjeto.partes.size() == mundo_Ancho * mundo_Alto){
                finalJuego1 = true;
                return;
            } else{
                colocarObjeto();
            }
            if(puntuacion % 100 == 0 && tick - tick_decremento > 0){
                tick -= tick_decremento;
            }
        }
    }
}
}
}

```

Clase Configuraciones

```

package com.metodologia.videojuegodav.juego;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import com.metodologia.videojuegodav.ArchivoES;

public class Configuraciones {

```

```

public static boolean soundEnabled = true;
public static int activarnieles=1;
public static int felicitacion=0;
public static int[] highscores = new int[] { 100, 80, 50, 30, 10 };
public static void load(ArchivoES files){
    BufferedReader in = null;
    try {
        in = new BufferedReader(new
            InputStreamReader(files.leerArchivo(".piratas")));
        for (int i = 0; i < 5; i++) {
            highscores [i] = Integer.parseInt(in.readLine());
        }
    } catch (IOException e) {
        // TODO: handle exception
    } catch (NumberFormatException e) {
        // TODO: handle exception
    } finally{
        try {
            if(in != null)
                in.close();
        } catch (IOException e) {
            // TODO: handle exception
        }
    }
}

```

```

public static void save(ArchivoES files){
    BufferedWriter out = null;
    try {
        out = new BufferedWriter(new
            OutputStreamWriter(files.escribirArchivo(".piratas")));
        out.write("\n");
        for (int i = 0; i < 5; i++) {
            out.write(Integer.toString(highscores[i]));
            out.write("\n");
        }
    } catch (IOException e) {
        // TODO: handle exception
    } finally{
        try {
            if(out != null)
                out.close();
        } catch (IOException e) {
            // TODO: handle exception
        }
    }
}

```

```

public static void addScore(int score){
    for (int i = 0; i < 5; i++) {
        if(highscores[i]< score){
            for (int j = 4; j > i; j--)

```

```

        highscores[j]= highscores[j - 1];
        highscores[i]=score;
        break;
    }
}
}
}

```

Clase MovimientoObjeto

```

package com.metodologiaa.videojuegodav.juego;
import java.util.ArrayList;
import java.util.List;

```

```

public class MovimientoObjeto {
    public static final int ARRIBA = 0;
    public static final int IZQUIERDA = 1;
    public static final int ABAJO = 2;
    public static final int DERECHA = 3;
    public List<Tripulacion> partes = new ArrayList<Tripulacion>();
    public int direccion;
    public MovimientoObjeto(){
        direccion = DERECHA;
        partes.add(new Tripulacion(0, 7));
    }

    public void girarIzquierda(){
        direccion +=1;
        if(direccion > DERECHA){
            direccion = ARRIBA;
        }
    }

    public void girarDerecha(){
        direccion -=1;
        if(direccion < ARRIBA){
            direccion = DERECHA;
        }
    }

    public void abordaje(){
        Tripulacion end = partes.get(partes.size() - 1);
        partes.add(new Tripulacion(end.x, end.y));
    }

    public void avance(){
        Tripulacion barco = partes.get(0);

        int len = partes.size() - 1;
        for (int i = len; i > 0; i--) {
            Tripulacion antes = partes.get(i - 1);
            Tripulacion parte = partes.get(i);
            parte.x = antes.x;
            parte.y = antes.y;
        }
        if(direccion == ARRIBA)
            barco.y -= 1;
        if(direccion == IZQUIERDA)

```

```

        barco.x -= 1;
    if(direccion == ABAJO)
        barco.y += 1;
    if(direccion == DERECHA)
        barco.x += 1;
    if(barco.x < 0)
        barco.x = 14;
    if(barco.x > 14)
        barco.x = 0;
    if(barco.y < 0)
        barco.y = 8;
    if(barco.y > 8)
        barco.y = 0;
}

public boolean comprobarChoque(){
    int len = partes.size();
    Tripulacion barco = partes.get(0);
    for (int i = 1; i < len; i++) {
        Tripulacion parte = partes.get(i);
        if(parte.x == barco.x && parte.y == barco.y)
            return true;
    }
    return false;
}

public boolean comprobarCaminomenor(int x1, int y1){
    int len = partes.size();
    Tripulacion barco = partes.get(0);
    if( barco.x <=x1 && barco.y <=y1)
        return true;
    return false;
}

public boolean comprobarCaminomayor(int x1, int y1){
    int len = partes.size();
    Tripulacion barco = partes.get(0);
    if( barco.x >=x1 && barco.y >=y1)
        return true;
    return false;
}

public boolean comprobarCaminointermedio(int x1, int y1, int x2, int y2){
    int len = partes.size();
    Tripulacion barco = partes.get(0);
    if( barco.x >=x1 && barco.x<=x2 && barco.y >= y1 && barco.y<=y2)
        return true;
    return false;
}
}

```

ANEXO IV

**PANTALLAS DEL VIDEOJUEGO “EL TREN DEL
CONOCIMIENTO”.**

PANTALLA SPLASH



PANTALLA MENÚ PRINCIPAL



PANTALLA NIVELES



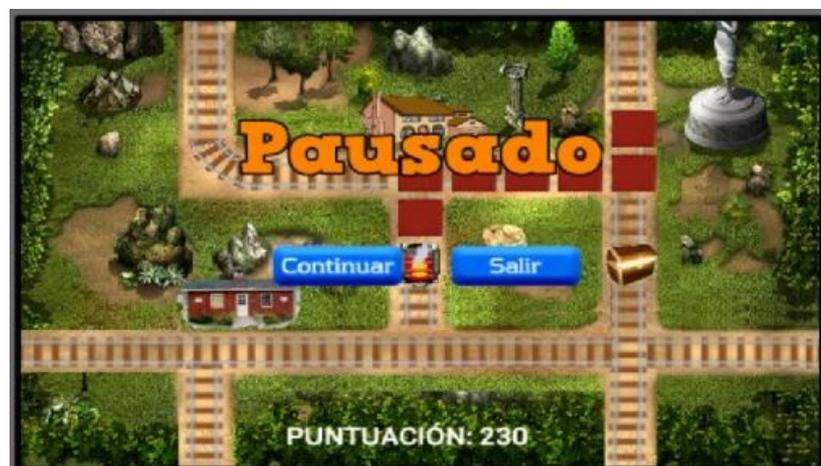
PANTALLA JUEGO “PREPARADO”



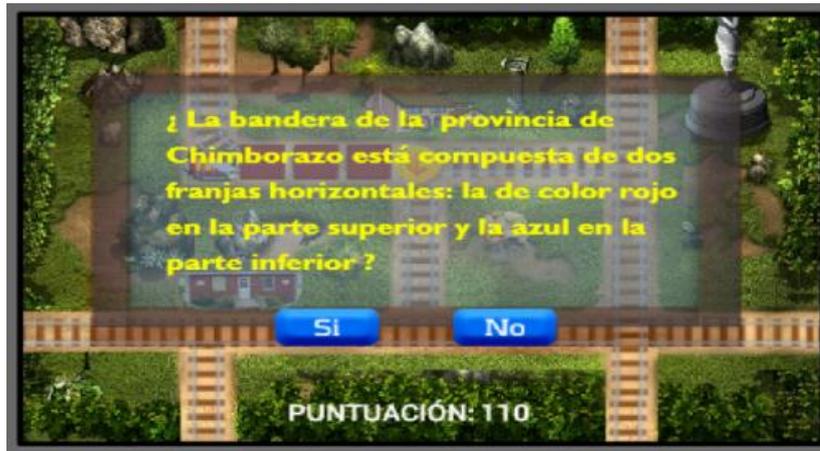
PANTALLA JUEGO “EJECUTANDOSE”



PANTALLA JUEGO “PAUSADO”



PANTALLA JUEGO “PREGUNTAS”



PANTALLA JUEGO “PREGUNTAS CORRECTAS”



PANTALLA JUEGO “PREGUNTAS INCORRECTAS”



PANTALLA JUEGO “FIN DEL JUEGO”



PANTALLA JUEGO “NIVEL COMPLETADO”



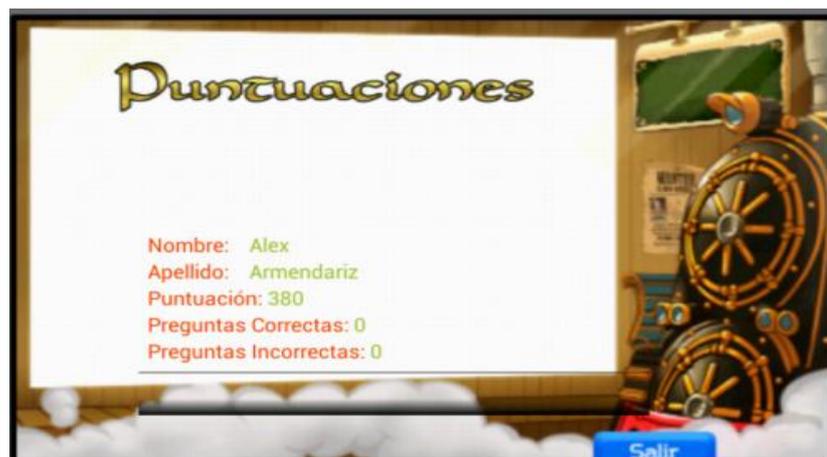
PANTALLA JUEGO “CONGRATULACIÓN”



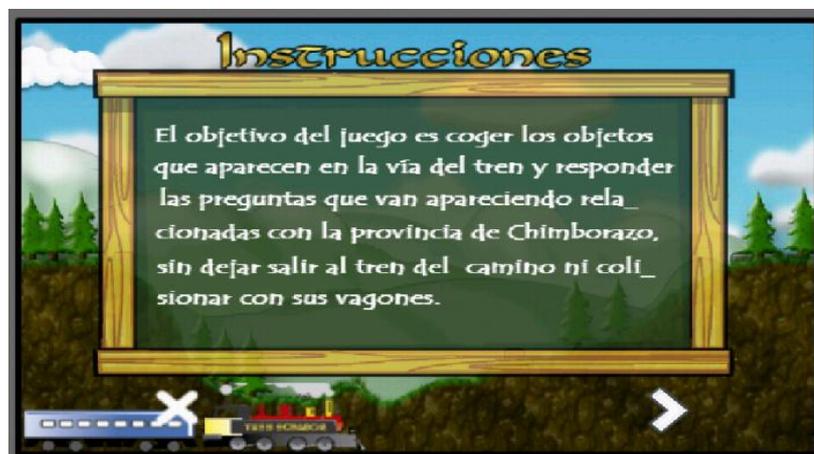
PANTALLA JUEGO “INGRESO DE DATOS DEL JUGADOR”



PANTALLA PUNTUACIONES



PANTALLA INSTRUCCIONES



PANTALLA CRÉDITOS



PANTALLA SALIR

